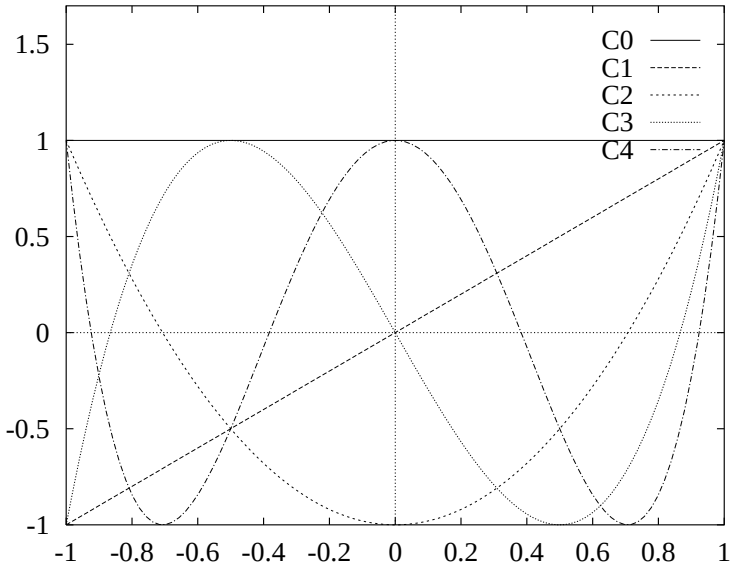
Daniel Ioan, Irina Munteanu



Metode Numerice

Algoritmi fundamentali şi aplicaţii în ingineria electrică

Editura
Bucureşti, 2000

Daniel Ioan, Irina Munteanu
Catedra de Electrotehnică, Universitatea "Politehnica" din Bucureşti

Metode Numerice

Algoritmi fundamentali şi aplicaţii în ingineria electrică

Editura
Bucureşti, 2000

Daniel Ioan, Irina Munteanu

Metode Numerice. Algoritmi fundamentali şi aplicaţii în ingineria electrică

Referenţi ştiinţifici: xxx
yyy

Editura, Bucureşti, 2000

Adresa editura

Cuprins

Capitolul 8

Ecuaţii diferenţiale ordinare

Ecuaţiile şi sistemele de ecuaţii diferenţiale ordinare au un rol central în modelarea fenomenelor fizice, chimice, sociologice, biologice sau de altă natură, caracterizate de un număr finit de mărimi de stare variabile în timp. Exemplul tipic în ingineria electrică îl reprezintă circuitele electrice în regim tranzitoriu, care sunt caracterizate de sisteme de astfel de ecuaţii. De altfel, teoria sistemelor cu o multitudine de aplicaţii în ingineria electrică, este bazată pe astfel de ecuaţii.

Forma canonică a sistemelor de ecuaţii diferenţiale este:

$$\left\{\begin{array}{rcl} \frac{dx_1}{dt} & = & f_1(x_1, x_2, \ldots, x_n, t) \\ \frac{dx_2}{dt} & = & f_2(x_1, x_2, \ldots, x_n, t) \\ \vdots & & \\ \frac{dx_n}{dt} & = & f_n(x_1, x_2, \ldots, x_n, t), \end{array}\right.$$

sau sub formă compactă:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \tag{8.1}$$

unde $\mathbf{f} : \mathbf{R}^{n+1} \to \mathbf{R}^n$ este o funcţie vectorială cu valori în spaţiul euclidian n-dimensional, $\mathbf{x} \in \mathbf{R}^n$ vectorul soluţie, funcţie de variabila scalară, $t \in \mathbf{R}$.

În continuare, atunci când acest lucru nu generează confuzii, vectorul soluţie va fi notat şi cu x, derivata cu x', iar funcţia cu f, astfel încât relaţia $x' = f(x, t)$ este echivalentă cu (8.1).

Trebuie remarcat că o ecuaţie diferenţială arbitrară de ordin superior: $x^{(n)} = f(x, x', \ldots, x^{(n-1)}, t)$ poate fi adusă la forma canonică prin simpla notaţie:

$$\left\{\begin{array}{l} x_i'(t) = x_{it}(t), \quad i = 1, 2, \ldots, n \\ f_n'(t) = f(t, x_1(t), x_2(t), \ldots, x_n(t)). \end{array}\right.$$

După cum se va constata ulterior, pentru ca acest sistem să aibă soluţie unică este necesară adăugarea unor condiţii suplimentare impuse soluţiei, numite condiţii iniţiale dacă se referă la o valoare particulară a variabilei t, sau condiţii **bilocale**, dacă se referă la două valori particulare ale variabilei independente.

8.1 Formularea corectă a problemelor cu condiţii iniţiale

Abordarea ecuaţiei (8.1) prin metode numerice este lipsită de sens, dacă problema nu este bine formulată din punct de vedere matematic. Pentru ecuaţiile care nu au soluţie sau au mai multe soluţii, chiar şi cele mai "performante" metode de rezolvare numerică nu fac altceva decât să genereze un şir de numere lipsite de semnificaţie.

Din acest motiv, analiza calitativă a ecuaţiei care trebuie rezolvată, din punctul de vedere al bunei formulări a problemei asociate, este esenţială şi trebuie efectuată înaintea rezolvării numerice. În formularea sa clasică, datorată lui Cauchy, problema reazolvării ecuaţiilor diferenţiale constă în determinarea funcţiei vectoriale $\mathbf{x} = \hat{\mathbf{x}}(t)$ de variabila reală t, $\hat{\mathbf{x}} : [t_0, t_m] \to \mathbf{R}^n$, funcţie continuă şi derivabilă care satisface ecuaţia (8.1) pe intervalul $[t_0, t_m]$ şi condiţia:

$$\hat{\mathbf{x}}(t_0) = \mathbf{x}_0, \tag{8.2}$$

cu $\mathbf{x}_0 \in \mathbf{R}^n$ dat, numită condiţie iniţială.

Condiţia ca problema lui Cauchy să aibă soluţie este dată de teorema lui Peano, care impune în principiu doar continuitatea funcţiei f în punctul (x_0, t_0).

Teorema 8.1 de existenţă a lui Peano

Dacă funcţia $f(\mathbf{x}, t)$ este continuă pe mulţimea:

$$D : ||\mathbf{x} - \mathbf{x}_0|| \leq a, \qquad t_0 \leq t \leq t_m,$$

şi care în plus satisface condiţia iniţială (8.2), atunci există cel puţin o funcţie $\hat{\mathbf{x}}(t)$, care satisface ecuaţia diferenţială (8.1) pe un interval nevid:

$$t_0 \leq t \leq t_{0+h} \leq t_m.$$

Teorema lui Peano nu garantează unicitatea soluţiei ci doar existenţa acesteia. Intervalul pe care este asigurată existenţa soluţiei nu este nemărginit ci are o lungime dată de:

$$h = \min\left(t_m - t_0, \frac{a}{M}\right), \tag{8.3}$$

unde M este o margine superioară a valorii absolute a lui $\mathbf{f}(\mathbf{x}, t)$ pe D. Din acest motiv, teorema lui Peano este o teoremă de existenţă locală.
Demonstraţia teoremei lui Peano este prezentată în [20], [30].
Pentru a obţine unicitatea soluţiei este necesară întărirea condiţiilor din teorema lui Peano, până la o condiţie de tip **Lipschitz**.

Definiţia 8.1 *O funcţie* $\mathbf{f}(\mathbf{x},t):\mathbf{R}^{n+1}\to\mathbf{R}$ *satisface condiţia Lipschitz faţă de* $\mathbf{x}$ *pe o mulţime închisă* $D\subset\mathbf{R}^{n+1}$*, dacă există o constantă* L*, astfel încât:*

$$||\mathbf{f}(\mathbf{x}',t)-\mathbf{f}(\mathbf{x}'',t)|| \;\le\; L||\mathbf{x}'-\mathbf{x}''||, \tag{8.4}$$

pentru orice $(\mathbf{x}',t)\in D$ *şi* $(\mathbf{x}'',t)\in D$*.*

Funcţiile care satisfac condiţia Lipschitz sunt funcţii continue, nu neapărat derivabile, dar a căror "viteză de creştere" este mărginită. Funcţiile cu derivata continuă satisfac condiţia Lipschitz dar nu şi reciproc. De exemplu, funcţiile continue, liniare pe porţiuni satisfac definiţia anterioară, dar nu sunt în mod necesar derivabile. Figura 8.1a reprezintă graficul unei funcţii Lipschitz reale de o singură variabilă reală iar figura 8.1b reprezintă o funcţie care nu satisface această condiţie deoarece panta tangentei în x_0 este nemărginită.

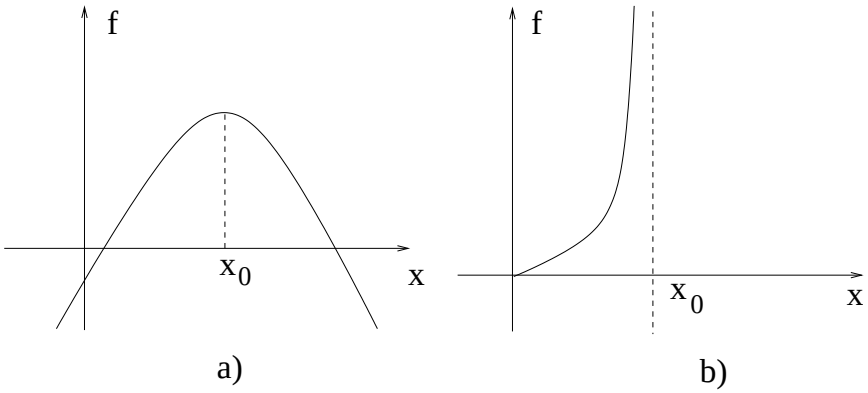


Figura 8.1: a) Funcţie Lipschitziană; b) Funcţie care nu satisface condiţia Lipschitz

Teorema 8.2 de existenţă şi unicitate a lui Picard

Dacă funcţia $\mathbf{f}(\mathbf{x},t)$ *este continuă pe mulţimea:*

$$D:\;||\mathbf{x}-\mathbf{x}_0||\;\le\;a,\quad t_0\le t\le t_m,$$

şi satisface condiţia Lipschitz în raport cu $\mathbf{x}$ *pe* D*, atunci există o funcţie unică* $\hat{\mathbf{x}}(t)$*, care satisface ecuaţia diferenţială (8.1) pe un interval nevid* $t_0\le t\le t_0+h\le t_m$ *şi condiţia iniţială (8.2).*

La fel ca teorema anterioară, şi teorema lui Picard are un caracter local.

Întărirea condiţiilor impuse funcţiei $\mathbf{f}(\mathbf{x},t)$ asigură nu numai unicitatea ci şi dependenţa continuă a soluţiei $\hat{\mathbf{x}}(t)$ de datele iniţiale $\mathbf{x}_0$ şi t_0.

Acest lucru este important mai ales când se urmăreşte obţinerea unei soluţii numerice, care porneşte de la date iniţiale obţinute pe cale experimentală, deci susceptibile de erori.

Demonstraţia toeremei de existenţă şi unicitate a lui Picard este prezentată în [20], [30].

Pentru a obţine existenţa soluţiei pe un interval de timp oricât de mare, este necesară impunerea unor condiţii şi mai severe pentru funcţia $\mathbf{f}(\mathbf{x},t)$, aşa cum rezultă din teorema următoare.

Teorema 8.3 de existenţă globală a lui Wintner

Dacă funcţia $\mathbf{f}(\mathbf{x},t)$ *este continuă pe întregul spaţiu* $(n+1)$*- dimensional* $\mathbf{R}^{n+1}$ *şi există o funcţie reală de variabilă reală* $L(r)$ *cu proprietatea* $\int_0^\infty \frac{dr}{L(r)} = \infty$, *astfel încât:*

$$|f_i(x_1, x_2, ..., x_n, t)| < L(r), \quad i = 1, 2, ..., n, \tag{8.5}$$

oricare ar fi $r = ||\mathbf{x}|| \in [0, \infty)$, *atunci soluţia ecuaţiei diferenţiale (8.1) cu condiţia (8.2) există pentru orice* $t \in \mathbf{R}$.

Teorema lui Wintner impune funcţiei $\mathbf{f}(\mathbf{x},t)$ o creştere moderată spre infinit. Cu toate că polinoamele de gradul doi sau mai mare nu satisfac această condiţie, totuşi funcţiile continue, liniare pe porţiuni, satisfac această condiţie, deci asigură existenţa globală a soluţiei.

În teoria circuitelor electrice, prezintă interes nu numai problemele cu condiţii iniţiale ci şi cele cu condiţii bilocale. Un exemplu de astfel de condiţie la limită o reprezintă condiţia de periodicitate:

$$\mathbf{x}(t_0) = \mathbf{x}(t_0 + T). \tag{8.6}$$

Din păcate, în cazul în care $\mathbf{f}(\mathbf{x},t)$ nu este liniară faţă de $\mathbf{x}$ această problemă nu are soluţie în general unică. În schimb, dacă integrala pe o perioadă a funcţiei $\mathbf{f}(\mathbf{x},t)$ satisface condiţia Lipschitz $\left\| \int_{t_0}^{t_0+T} [\mathbf{f}(\mathbf{x}',t) - \mathbf{f}(\mathbf{x}'',t)]dt \right\| \leq L||\mathbf{x}' - \mathbf{x}''||$ cu constanta $L < 1$, atunci soluţia ecuaţiei (8.1) cu condiţia (8.2) este unică.

8.2 Discretizarea ecuaţiilor diferenţiale

Determinarea numerică a soluţiei ecuaţiei diferenţiale (8.1) cu condiţia iniţială (8.2) presupune **discretizarea intervalului** $[t_0, t_m]$ cu paşi de timp relativ mici:

$$t_0, \quad t_1 = t_0 + h_0, \quad t_2 = t_1 + h_1, \quad ..., \quad t_{k+1} = t_k + h_k, \quad ..., \quad t_m.$$

Dacă pasul de timp are valoare constantă $h_k = h$, atunci se obţine o reţea uniformă de discretizare:

$$t_0, \quad t_1 = t_0 + h, \quad t_2 = t_0 + 2h, \quad ..., \quad t_{k+1} = t_0 + (k+1)h, \ ..., \ t_n = t_0 + nh.$$

O metodă numerică de rezolvare reprezintă un algoritm care permite determinarea numerică a unei aproximări a soluţiei în nodurile reţelei de discretizare:

$$\hat{\mathbf{x}}(t_0), \ \hat{\mathbf{x}}(t_1), \ \hat{\mathbf{x}}(t_2), \ ..., \ \hat{\mathbf{x}}(t_{k+1}), \ ..., \ \hat{\mathbf{x}}(t_m).$$

Deoarece valorile soluţiei numerice nu sunt riguros egale cu soluţia exactă, ele vor fi notate cu:

$$\mathbf{x}_0,\ \mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_k,\ \ldots,\ \mathbf{x}_m.$$

Abaterile soluţiei numerice $\mathbf{x}_k$ de la pasul de timp t_k, faţă de soluţia exactă $\mathbf{x}(t_k)$ la acelaşi moment de timp pot fi caracterizate prin eroarea absolută:

$$\varepsilon_k = ||\hat{\mathbf{x}}(t_k) - \mathbf{x}_k||. \tag{8.7}$$

Această eroare este datorată, în principal a două cauze:

- aproximarea care stă la baza algoritmului, care determină o componentă a erorii numită eroare **de trunchiere**;
- faptul că în calculator numerele reale sunt reprezentate prin aproximări raţionale, cu un număr finit de cifre semnificative, ceea ce determină erori numite **de rotunjire**.

Prima categorie de erori depinde de metoda numerică folosită iar a doua categorie este dependentă de sistemul de calcul utilizat.

În alegerea algoritmilor se urmăreşte ca erorile de orice natură să fie cât mai mici. Majoritatea algoritmilor asigură o scădere a erorii de trunchiere pe măsură ce pasul de discretizare a timpului, numit şi pas de integrare, scade. Proprietatea unui algoritm de a asigura limita nulă a erorii de trunchiere $\varepsilon_k \to 0$ pentru orice $k = 1, 2, \ldots, m$, atunci când $h \to 0$ poartă numele de **convergenţă**. În această definiţie se presupune că soluţia numerică îndeplineşte exact condiţia iniţială $(\varepsilon_0 = ||\hat{\mathbf{x}}(t_0) - \mathbf{x}_0|| = 0)$ şi h=max(h_k).

Modul în care depinde eroarea de trunchiere de mărimea pasului poate fi caracterizat de *ordinul metodei*. Dacă există un număr p, real şi pozitiv astfel încât eroarea de trunchiere este majorată de:

$$\varepsilon_k \leq Ch^p, \tag{8.8}$$

în care constanta C este independentă de pasul de timp h, atunci se spune că **metoda este de ordinul** p şi se scrie că $\varepsilon_k = O(h^p)$. Este de aşteptat ca metodele de ordin superior să asigure o eroare mai mică decât cele de ordin inferior, atunci când pasul de timp se micşorează.

Datorită caracterului recursiv al metodelor numerice de integrare a ecuaţiilor diferenţiale, în care $\mathbf{x}_k$ este calculat de regulă în funcţie de $\mathbf{x}_{k-1}$ erorile apărute la pasul k se propagă la paşii ulteriori, afectând întreaga soluţie. Pentru a studia propagarea erorii în procesul de calcul se consideră soluţia numerică $\mathbf{z}_k$ a ecuaţiei diferenţiale cu condiţia iniţală $\mathbf{z}_0$, perturbată astfel încât: $||\mathbf{x}_0 - \mathbf{z}_0|| = \varepsilon_0$.

O metodă de rezolvare se numeşte **numeric stabilă**, pentru un pas de timp h, dacă abaterea dintre cele două soluţii este mărginită:

$$||\mathbf{x}_k - \mathbf{z}_k|| = C\varepsilon_0, \tag{8.9}$$

în caz contrar metoda se numeşte *instabilă numeric*.

Dacă o metodă numerică determină "amplificarea" erorii în procesul de calcul, atunci ea nu poate fi utilizată practic în vederea obţinerii unor soluţii numerice cu eroare rezonabilă.

O metodă de rezolvare se numeşte **numeric absolut stabilă** pentru un pas de timp dat h şi pentru o ecuaţie dată, dacă o perturbaţie la pasul de timp k : $||\mathbf{x}_k - \mathbf{z}_k|| = \delta_k$, nu se amplifică la paşii de timp următori $m > k$:

$$||\mathbf{x}_m - \mathbf{z}_m|| \leq \delta_k. \tag{8.10}$$

Din păcate, definiţia stabilităţii numerice nu depinde numai de algoritm ci şi de ecuaţia de rezolvat. Din acest motiv este necesară introducerea unei "ecuaţii test" pe care să se verifice stabilitatea absolută. Se alege ca ecuaţie test cea mai simplă ecuaţie diferenţială liniară scalară:

$$\frac{dx}{dt} = \lambda x, \tag{8.11}$$

în care $\lambda \in \mathbf{C}$, deoarece multe sisteme liniare şi chiar prima aproximaţie a celor neliniare se pot reduce la ecuaţii de acest tip, printr-o schimbare convenabilă de variabile. Se va numi **regiune de stabilitate numerică absolută** mulţimea valorilor pe care le poate lua pasul de timp h, la o valoare dată a parametrului λ, astfel încât o perturbaţie a soluţiei numerice la pasul k va produce modificări ulterioare care nu cresc de la pas la pas. Un exemplu de astfel de perturbări îl constituie imprecizia cu care sunt cunoscute datele iniţiale sau erorile de rotunjire care intervin la fiecare pas de integrare în timp.

Pe măsură ce pasul de timp h scade, calculele sunt efectuate cu valori din ce în ce mai apropiate ($x_{k+1} \to x_k$) iar ponderea erorilor de rotunjire creşte. Efectul cumulativ al erorilor de trunchiere şi rotunjire este prezentat calitativ în figura 8.2. Se constată că în realitate pasul de timp nu trebuie să scadă sub o valoare minimă h_{min} deoarece erorile de rotunjire tind să crească din nou.

În concluzie, se poate afirma că prezintă interes practic doar acele metode care sunt convergente şi numeric stabile, cu condiţia ca pasul de integrare să nu fie mai mic decât limita impusă de eroarea de rotunjire.

În continuare va fi studiată cea mai simplă metodă de discretizare (de integrare numerică) a ecuaţiilor diferenţiale ordinare, şi anume metoda de ordinul unu, cunoscută şi sub numele de **metoda Euler**. Aceasta se obţine prin aproximarea soluţiei exacte cu primii doi termeni din seria Taylor în care aceasta se poate descompune în fiecare punct al unei reţele uniforme de discretizare:

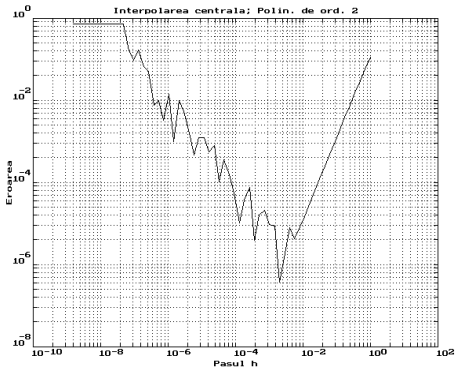


Figura 8.2: Variaţia erorii globale în funcţie de pasul h

$$\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + (t_{k+1} - t_k) \left.\frac{d\mathbf{x}}{dt}\right|_{t=t_k} + \dots \tag{8.12}$$

sau ţinând seama de ecuaţia de rezolvat şi de notaţiile introduse, rezultă:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}(\mathbf{x}_k, t_k). \tag{8.13}$$

În fond, această relaţie se obţine din ecuaţia de rezolvat aproximând derivata cu formula progresivă cu diferenţe finite de ordinul întâi. Relaţia (8.13) defineşte o **metodă Euler explicită**, deoarece permite calculul direct al soluţiei numerice $\mathbf{x}_{k+1}$, fără să fie necesară rezolvarea vreunei ecuaţii sau sistem de ecuaţii.

Relaţia (8.13) poate fi obţinută şi prin integrarea ecuaţiei diferenţiale (8.1) pe ultimul pas de timp

$$\hat{\mathbf{x}}(t_{k+1}) - \hat{\mathbf{x}}(t_k) = \int_{t_k}^{t_{k+1}} \mathbf{f}(\mathbf{x}, t)dt, \tag{8.14}$$

şi aproximarea integralei prin metoda dreptunghiurilor cu valoarea integrandului din momentul iniţial t_k, al intervalului.

Dacă, în schimb, se foloseşte pentru evaluarea integralei valoarea din momentul final t_{k+1}, atunci se obţine relaţia:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) \tag{8.15}$$

caracteristică **metodei Euler implicite**. Această relaţie se obţine aproximând derivata din ecuaţia de rezolvat cu diferenţe finite regresive. În acest ultim caz, necunoscuta $\mathbf{x}_{k+1}$, este prezentă atât în membrul stâng cât şi în cel drept iar pentru determinarea ei este necesară rezolvarea unui sistem de ecuaţii neliniare. Efortul de calcul în cazul metodelor implicite este mai mare, dar se va constata că el este recompensat de o stabilitate numerică superioară a soluţiei.

Semnificaţia geometrică a metodei Euler explicită este reprezentată în figura 8.3, pentru cazul unei funcţii $f(x,t)$ cu $x \in \mathbf{R}$. Se constată că soluţia este aproximată cu o dreaptă care trece prin punctul iniţial (x_k, t_k) şi a cărei pantă este dată de valoarea funcţiei $f(x,t)$ în punctul iniţial. Pe măsură ce pasul de timp h scade, are loc şi o scădere a erorii de trunchiere ε_k.

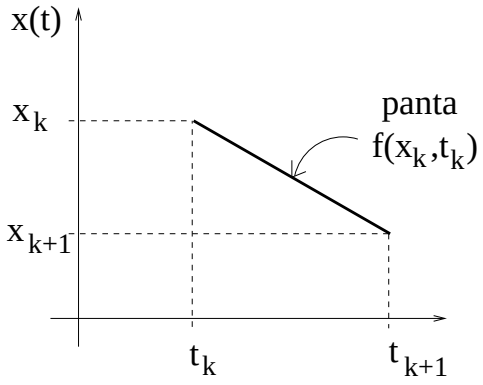


Figura 8.3: Semnificaţia geometrică a metodei Euler explicite

Pentru a determina **ordinul de convergenţă** al metodei se consideră ecuaţia test (8.11) cu $\lambda \in \mathbf{R}$, a cărei soluţie este:

$$\hat{x}(t) = x_0 e^{\lambda t},$$

şi care, în nodurile $x_k = kh$ ale unei reţele regulate are valorile:

$$\hat{x}(t_k) = x_0 e^{\lambda h t}.$$

Soluţia numerică se obţine folosind recursiv relaţia (8.13):

$$x_{k+1} = x_k + h\lambda x_k = x_0(1+\lambda h)^{k+1}.$$

Dacă se notează:

$$\alpha = e^{\lambda h} = 1 + \lambda h + \frac{(\lambda h)^2}{2} e^{\theta \lambda h}, \qquad 0 < \theta < 1 \quad \text{şi} \tag{8.16}$$

$$\beta = 1 + \lambda h \leq \alpha,$$

rezultă următoarele expresii pentru soluţia exactă şi cea numerică:

$$\hat{x}(t_k) = x_0 \alpha^k, \tag{8.17}$$

$$x_k = x_0 \beta^k. \tag{8.18}$$

Eroarea de trunchiere are la pasul k valoarea:

$$\varepsilon_k = |\hat{x}(t_k) - x_k| = x_0 |\alpha^k - \beta^k| = x_0 |\alpha - \beta| \; |\alpha^{k-1} + \beta\alpha^{k-2} + ... + \beta^{k-1}|.$$

Luând în consideraţie (8.16),rezultă:

$$\varepsilon_k \leq x_0 \frac{(\lambda h)^2}{2} e^{\theta \lambda h} k \alpha^{k-1} \leq x_0 h^2 k \lambda^2 e^{\lambda k h}, \tag{8.19}$$

Fixând valoarea lui $t_k = kh$, rezultă evaluarea:

$$e_k \leq h x_0 \lambda^2 t_k e^{\lambda t_k} = O(h), \tag{8.20}$$

care evidenţiază faptul că metoda Euler este convergentă cu ordinul 1. Relaţia (8.20) evaluează eroarea cumulată pe toţi paşii de timp până la t_k, motiv pentru care ea se numeşte **eroare globală**. Eroarea de trunchiere efectuată la fiecare pas, presupunând valoarea soluţiei la pasul anterior ca exactă, se numeşte **eroare locală** şi se poate calcula folosind relaţia (8.20) în care se particularizează $k = 1$:

$$e_1 \leq x_0 h^2 \lambda^2 e^{\lambda h} = O(h^2). \tag{8.21}$$

Se constată că **eroarea locală** este cu un ordin de mărime mai mare decât cea globală.

Presupunând că funcţia $f(x,t)$ satisface condiţia Lipschitz şi că soluţia are derivata dublă mărginită, se obţin [47] aceleaşi rezultate privind ordinul erorii metodei Euler şi în cazul general al ecuaţiei (8.1).

Pentru a studia **stabilitatea numerică** a metodei Euler se consideră soluţia ecuaţiei de test (8.11) şi soluţia ecuaţiei cu condiţii iniţiale perturbate:

$$z_k = z_0 \beta^k.$$

Abaterea soluţiei perturbate:

$$|x_k - z_k| = |x_0 - z_0| \beta^k \leq \varepsilon_0 \alpha^k = \varepsilon_0 e^{\lambda h k} = \varepsilon_0 e^{\lambda t_k} \tag{8.22}$$

este mărginită de abaterea condiţiei iniţiale $\varepsilon_0 = |x_0 - z_0|$ şi tinde către 0 atunci când $\varepsilon_0 \to 0$.

În consecinţă, se poate afirma că metoda Euler este numeric stabilă pentru orice valoare a pasului de integrare. Metoda este numeric stabilă nu numai pentru ecuaţiile stabile ($Re[\lambda] < 0$) ci şi pentru cele instabile, a căror soluţie nu este asimptotic mărginită. Această afirmaţie este adevărată şi în cazul ecuaţiei diferenţiale (8.1). În acest caz [26], abaterea celor două soluţii este mărginită de:

$$||\mathbf{x}_k - \mathbf{z}_k|| \leq \varepsilon_0 e^{Lh},$$

unde L este constanta Lipschitz a funcţiei $\mathbf{f}(\mathbf{x}, t)$.

Stabilitatea numerică a unei metode nu presupune convergenţa acesteia. De exemplu "metoda de ordin 0", obţinută prin reţinerea primului termen din seria Taylor (8.13):

$$\mathbf{x}_1 = \mathbf{x}_0,\ \mathbf{x}_2 = \mathbf{x}_1,\ \mathbf{x}_3 = \mathbf{x}_2,\ \ldots,\ \mathbf{x}_{k+1} = \mathbf{x}_k,\ \ldots$$

este numeric stabilă (chiar şi absolut stabilă) dar nu este convergentă.

Pentru a asigura **stabilitatea numerică absolută** este necesar ca în $(\mathbf{x_1})$ să fie îndeplinită condiţia:

$$|x_k - z_k| \leq |x_0 - z_0|,$$

sau echivalent

$$|\beta|^k = |1 + \lambda h|^k \leq 1, \tag{8.23}$$

ceea ce corespunde în planul complex λh unui cerc de rază unitate, cu centrul în $(-1, 0)$.

Acest cerc (figura 8.4) reprezintă domeniul de stabilitate numerică absolută pentru metoda Euler.

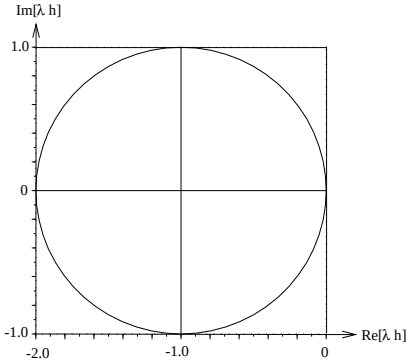


Figura 8.4: Domeniul de stabilitate numerică absolută pentru o metodă numerică

Dacă se consideră spre exemplu ecuaţia unui circuit electric liniar de ordinul 1, cu constanta de timp τ, care este de forma ecuaţiei de test (8.11) cu $\lambda = -1/\tau$. În acest caz condiţia de stabilitate numerică absolută este $-2 \leq \lambda h \leq 0$. Dacă se consideră pasul de timp limită $h = -\frac{2}{\lambda} = 2\tau$, atunci soluţia numerică obţinută prin metoda Euler este

$$x_k = (1 + \lambda h)x_{k-1} = (-1)^k x_0.$$

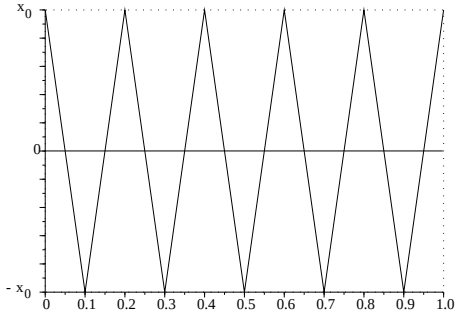


Figura 8.5: Soluţia numerică Euler a unui circuit liniar de ordinul 1, pentru $h = 2\tau$

Aceasta are un caracter oscilator (figura 8.5), în timp ce soluţia exactă tinde către zero în câţiva paşi de timp. Dacă pasul de timp depăşeşte această limită, atunci oscilaţiile se amplifică, soluţia numerică devine instabilă şi se îndepărtează cu fiecare pas de integrare de soluţia exactă a ecuaţiei de rezolvat. Pentru a obţine o variaţie monotonă a soluţiei numerice, aşa cum variază soluţia exactă, este necesar ca pasul de integrare h să fie mai mic decât constanta de timp τ a circuitului.

Pentru a caracteriza **eroarea de rotunjire** se consideră un sistem de calcul care operează cu q cifre semnificative. La fiecare operaţie acesta introduce o eroare relativă de rotunjire $r \in [-1/2, 1/2] \cdot 10^{-q}$. Dacă pe acest sistem se implementează metoda Euler pentru ecuaţia (8.11) cu $x_0 = 1$ şi $\lambda \equiv -1/2$, atunci eroarea de rotunjire la pasul $t_k = \tau$ este:

$$\varepsilon_r = Nr \cong \frac{\tau}{h} 10^{-q},$$

deoarece au fost efectuaţi $N = \tau/h$ paşi. Eroarea de trunchiere la acelaşi pas se obţine pe baza relaţiilor (8.19), (8.20):

$$\varepsilon_t = \frac{h}{2\tau^2} \tau e^{-1} \cong \frac{h}{\tau}.$$

Suma acestor erori:

$$\varepsilon = \varepsilon_t + \varepsilon_r = \frac{h}{\tau} + \frac{\tau}{h} 10^{-q},$$

are minimul dat de ecuaţia:

$$\frac{\partial \varepsilon}{\partial h} \equiv \frac{1}{\tau} - \frac{\tau}{h^2} 10^{-q} = 0,$$

ceea ce corespunde unei valori optime a pasului de timp:

$$h = \tau \sqrt{10^{-q}}.$$

De exemplu, pentru cazul uzual în care $q = 6$, rezultă că pasul metodei Euler trebuie ales mai mare de 1/1000 din constanta de timp maximă din circuit dar mai mic decât constanta de timp minimă din circuit.

Rezolvarea prin metoda Euler explicită este descrisă de următorul pseudocod:

```
procedura euler_exp (x0, xmax, y0, h, y)
        real x0,                ;nod iniţial
             xmax,              ;nod final
             y0,                ;condiţia iniţială
             h                  ;pasul de integrare
        tablou real y(N)        ;vectorul soluţiei
                                ;numerice
        t = x0
        y(1) = y0
        n = (xmax - x0)/h       ;nr.  de paşi
```

```
            pentru i = 2,n          ;evaluează funcţia
                                    ;în cele n noduri
                    y(i) = y(i-1) + hfunc(t,y(i-1))
                    t = t + h
    retur
```

Procedura are următorii parametri :

- de intrare
  - `x0` = limita inferioară a intervalului de integrare;
  - `xmax` = limita superioară a intervalului de integrare;
  - `y0` = condiţia iniţială;
  - `h` = pasul de integrare.
- de ieşire
  - `y(N)` = vectorul soluţie.

Procedura apelează funcţia "`func`" ce evaluează funcţia $f(t,y)$ din membrul drept al ecuaţiei (8.1) pentru nodul i.

Rezolvarea prin metoda Euler implicită este descrisă de următorul pseudocod:

```
procedura euler_imp (x0, xmax, y0, h, err, itmax, y)
        real x0,                ;nod iniţial
             xmax,              ;nod final
             y0,                ;condiţia iniţială
             h,                 ;pasul de integrare
             err                ;eroarea maxim admisă
        întreg itmax            ;nr. max. de iteraţii
        tablou real y(N)        ;vectorul soluţiei

        t = x0
        y(1) = y0
        n = (xmax - x0)/h       ;nr. de paşi
        pentru i = 2,n          ;evaluează funcţia
                                ;în cele n noduri
                t = t + h       ;pas nou
                ;prima evaluare se face cu metoda explicită
                ynou = y(i-1)+hfunc(t,y(i-1))

                j = 0           ;contor iteraţii
                repetă
```

```
                    yvechi = ynou
                    ynou = y + hfunc(t,yvechi)
                            ;evaluare nouă
                    j = j + 1
                    eps = abs(yvechi - ynou)
                            ;evaluare eroare
            până când ( abs(eps) ≤ err sau j > itmax )
            y(i) = ynou
retur
```

Faţă de metoda Euler explicită, procedura de mai sus are doi parametri suplimentari:

- de intrare
  - `err` = eroarea maximă admisă;
  - itmax = nr. maxim de iteraţii admis.

Deci, metoda de rezolvare a ecuaţiei neliniare (8.5) este o metodă iterativă, la care iniţializarea este făcută prin valori date de relaţia explicită (8.3).

În figurile 8.6 – 8.8 sunt prezentate soluţia exactă şi soluţiile numerice prin metodele Euler explicită şi implicită, la rezolvarea ecuaţiei diferenţiale a unui circuit RC serie:

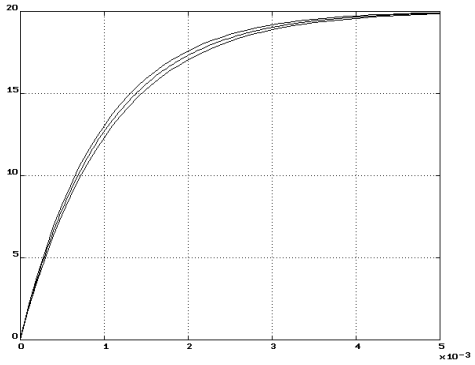


Figura 8.6: Tensiunea la bornele condensatorului pentru $h = \tau/10$

$$\begin{cases} RC\frac{\mathrm{d}u}{\mathrm{d}t} + u = E, \\ u(0) = 0. \end{cases}$$

pentru valorile pasului de discretizare temporală $h = \tau/10$, $h = \tau$ şi respectiv $h = 2\tau$.

Se constată că metoda Euler implicită este mult mai robustă, fiind mai puţin sensibilă la mărimea pasului.

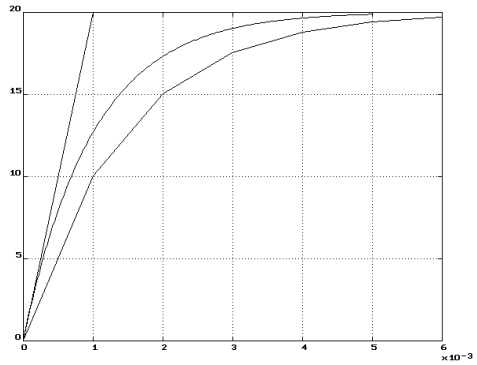


Figura 8.7: Tensiunea la bornele condensatorului pentru $h = \tau$

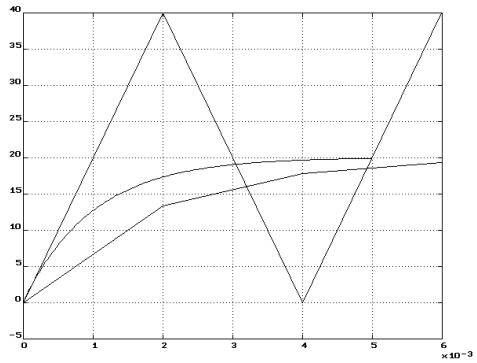


Figura 8.8: Tensiunea la bornele condensatorului pentru $h = 2\tau$

Un alt fenomen interesant este acela că, pentru o valoare suficient de mică a pasului, soluţia exactă este încadrată de cele două soluţii numerice (soluţia obţinută prin metoda Euler explicită fiind întotdeauna mai mare, iar cea obţinută prin metoda Euler implicită fiind întotdeauna mai mică decât cea exactă). Această observaţie permite rezolvarea unei ecuaţii diferenţiale ordinare cu control al erorii, prin utilizarea **ambelor** metode.

8.3 Metode numerice cu un pas

8.3.1 Metoda seriei Taylor

Pentru a obţine îmbunătăţirea convergenţei faţă de metoda Euler, se folosesc metode în care din seria Taylor a soluţiei se reţin mai mult de doi termeni. În urma trunchierii seriei Taylor la primii $(p+1)$ termeni se obţine relaţia:

$$x_{k+1} = x_k + hf(x_k, t_k) + \frac{h^2}{2!} f^{(1)}(x_k, t_k) + ... + \frac{h^p}{p!} f^{(p-1)}(x_k, t_k). \tag{8.24}$$

Dacă se aplică această metodă ecuaţiei test (8.11) cu $\lambda \in \mathbf{R}$, atunci soluţia numerică se

obţine de forma:

$$x_k = x_0\left(1 + \lambda h + \frac{(\lambda h)^2}{2!} + ... + \frac{(\lambda h)^p}{p!}\right)^k. \tag{8.25}$$

Folosind notaţiile:

$$\beta = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + ... + \frac{(\lambda h)^p}{p!};$$

$$\alpha = e^{\lambda h} = \beta + \frac{(\lambda h)^{p+1}}{p!} e\theta\lambda h \geq \beta;$$

în care $\theta \in (0,1)$, rezultă următoarele expresii pentru soluţia exactă şi cea numerică:

$$\hat{x}(t_k) = x_0\alpha^k, \tag{8.26}$$

$$x_k = x_0\beta^k. \tag{8.27}$$

Eroarea de trunchiere satisface majorările:

$$\begin{aligned}\varepsilon_k = |\hat{x}(t_k) - x_k| = x_0|\alpha^k - \beta^k| \leq x_0|\alpha - \beta|k\alpha^{k-1} \leq \\ \leq \frac{x_0(\lambda h)^{p+1}}{p!} e^{\theta\lambda h} k e^{\lambda h(k-1)} \leq h^p x_0 |\lambda|^p t_k e^{\lambda t_k}.\end{aligned} \tag{8.28}$$

Rezultă că această metodă este convergentă şi are ordinul p, eroarea de trunchiere fiind de tipul:

$$\varepsilon_k = O(h^p).$$

În această metodă, micşorarea erorii se poate realiza atât prin micşorarea pasului de timp h cât şi prin mărirea ordinului p al metodei. Micşorarea excesivă a pasului h duce la creşterea erorii de rotunjire; creşterea ordinului la valori mai mari de 1 presupune calculul numeric sau analitic al valorilor funcţiei $f(x,t)$, fapt ce sporeşte sensibil efortul de calcul şi este susceptibil de erori suplimentare. Din acest motiv metoda seriei Taylor trunchiată este folosită în practică doar pentru ordinul $p = 1$.

8.3.2 Metoda Runge-Kutta

O clasă de metode de ordin superior, care asigură aceeaşi comportare a erorii ca metoda Taylor, dar nu pretinde calculul derivatelor funcţiei $f(x,t)$ este cunoscută sub numele Runge- Kutta. Pentru a asigura acelaşi ordin al erorii, metodele Runge-Kutta evaluează funcţia $f(x,t)$ de mai multe ori, într-un număr p de puncte în vecinătatea punctului iniţial (x_k, t_k).

Metoda Runge-Kutta de ordinul doi utilizează relaţia de recurenţă:

$$x_{k+1} = x_k + h[a_1 f(x_k, t_k) + a_2 f(x_k + b_1 h f(x_k, t_k), t_k + b_2 h)], \tag{8.29}$$

în care, constantele a_1, a_2, b_1 şi b_2 se determină astfel încât eroarea de trunchiere să fie de tipul $O(h^2)$. Dacă se notează:

$$g_1 = f(x_k, t_k),$$
$$g_2 = f(x_k + b_1 h g_1, t_k + b_2 h),$$

relaţia (8.29) devine:

$$x_{k+1} = x_k + h[a_1 g_1 + a_2 g_2].$$

Derivând funcţia $f(x,t)$ faţă de timp:

$$f^{(1)}(x_k, t_k) = \frac{\partial f}{\partial x} f(x_k, t_k) + \frac{\partial f}{\partial t}, \tag{8.30}$$

rezultă prin reţinerea primilor termeni ai dezvoltării în serie Taylor a funcţiei g_2:

$$g_2 = f(x_k, t_k) + \frac{\partial f}{\partial x} f(x_k, t_k) b_1 h + \frac{\partial f}{\partial t} b_2 h + O(h^2),$$

iar prin substituţie în (8.29), rezultă:

$$x_{k+1} = x_k + h(a_1 + a_2) g_1 + h^2 a_2 b_1 \frac{\partial f}{\partial x} g_1 + h^2 a_2 b_2 \frac{\partial f}{\partial t} + O(h^3). \tag{8.31}$$

Identificând termenii expresiei (8.31) cu cei ai expresiei (8.24) particularizată pentru $p = 2$:

$$x_{k+1} = x_k + h g_1 + \frac{h^2}{2} \left[\frac{\partial f}{\partial x} g_1 + \frac{\partial f}{\partial t} \right],$$

rezultă sistemul de ecuaţii:

$$a_1 + a_2 = 1; \quad a_2 b_1 = 1/2; \quad a_2 b_2 = 1/2. \tag{8.32}$$

Cele trei relaţii obţinute nu pot determina în mod unic cei patru parametri şi unul dintre aceştia trebuie ales arbitrar. Indiferent de această alegere, metoda obţinută este convergentă cu o eroare de trunchiere de ordinul doi $\varepsilon_k = O(h^2)$. Se constată că în metoda Runge -Kutta de ordinul doi, la fiecare pas de timp se evaluează funcţia $f(x,t)$ de două ori, o dată în punctul iniţial (g_1) şi apoi într-un punct intermediar (g_2), plasat pe direcţia g_1, urmând ca valoarea acceptată pentru soluţia numerică să medieze ponderat cele două evaluări.

În practică se folosesc mai des următoarele variante ale metodei Runge-Kutta de ordinul doi:

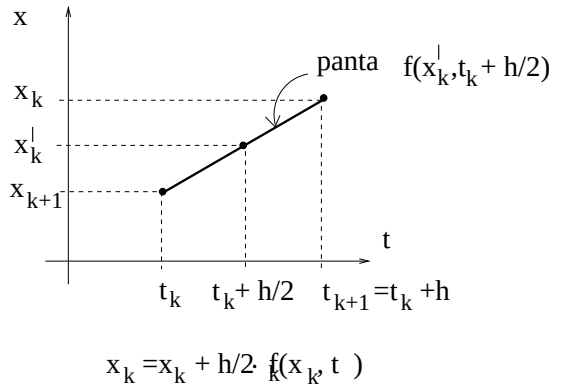


Figura 8.9: Semnificaţia geometrică a metodei Euler – Cauchy

- **metoda Euler-Cauchy**, obţinută pentru $a_2 = 1$:

$$x_{k+1} = x_k + hf\left(x_k + \frac{h}{2}f(x_k, t_k), t_k + \frac{h}{2}\right), \tag{8.33}$$

în care se evaluează funcţia $f(x,t)$ la jumătatea pasului de timp, într-un punct median, de pe segmentul de pantă $f(x_k, t_k)$, ce porneşte din punctul iniţial (semnificaţia geometrică este prezentată în figura 8.9);

- **metoda Henn**, obţinută pentru $a_2 = 1/2$:

$$x_{k+1} = x_k + \frac{h}{2}[f(x_k, t_k) + f(x_k + hf(x_k, t_k), t_k + h)], \tag{8.34}$$

în care se utilizează media aritmetică a valorilor funcţiei $f(x,t)$ în punctul iniţial (x_k, t_k) şi cel final $(x_k + hf(x_k, t_k), t_k + h)$, în sensul indicat de metoda Euler (semnificaţia geometrică este prezentată în figura 8.10);

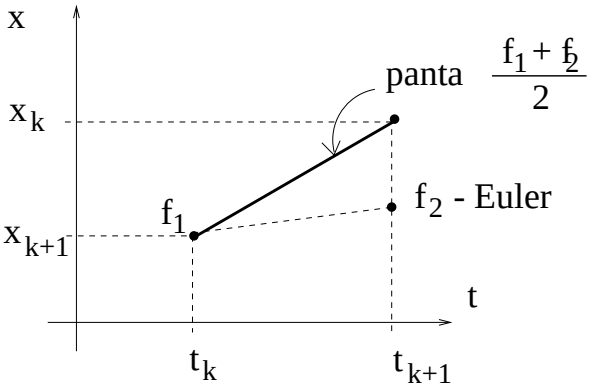


Figura 8.10: Semnificaţia geometrică a metodei Henn

- **metoda Ralston**, obţinută pentru $a_2 = 3/4$:

$$x_{k+1} = x_k + \frac{h}{4}\left[f(x_k, t_k) + 3f\left(x_k + \frac{2}{3}hf(x_k, t_k), t_k + \frac{2}{3}h\right)\right], \tag{8.35}$$

în care se utilizează o medie ponderată a valorii iniţiale (pondere 1/4) şi a valorii din punctul situat la 2/3 din pasul h (pondere 3/4). Această alegere asigură minimul erorii de trunchiere [26].

Folosind o tehnică asemănătoare se obţine metoda Runge-Kutta de **ordinul trei** cu o eroare de trunchiere de tipul $O(h^3)$ şi necesită trei evaluări ale funcţiei $f(x,t)$ la fiecare pas de timp. În acest caz numărul coeficienţilor se ridică la opt iar numărul ecuaţiilor obţinute prin identificarea cu seria Taylor este doar de şase, deci doi parametri rămân nedeterminaţi [2], [26].

Dintre toate variantele metodei Runge-Kutta, cea mai des folosită în practică este cea de **ordinul patru**:

$$x_{k+1} = x_k + h[a_1 g_1 + a_2 g_2 + a_3 g_3 + a_4 g_4], \tag{8.36}$$

cu

$$\begin{aligned} g_1 &= f(x_k, t_k); \\ g_2 &= f(x_k + b_1 g_1 h, t_k + c_1 h); \\ g_3 &= f(x_k + b_2 g_1^h + b_3 g_2^h, t_k + c_2 h); \\ g_4 &= f(x_k + b_4 g_1^h + b_5 g_2^h + b_6 g_3^h, t_k + c_3 h). \end{aligned}$$

În urma dezvoltării în serie Taylor a funcţiilor ce intervin în expresia (8.36) şi a identificării cu relaţia (8.24), particularizată pentru $p = 4$, rezultă un sistem de 11 ecuaţii satisfăcute de cei 13 coeficienţi $a_1 - a_4,\ b_1 - b_6,\ c_1 - c_4$.

Cel mai des utilizată este varianta Runge, în care se alege $b_6 = 1$ şi $c_1 = 1/2$:

$$x_{k+1} = x_k + h(g_1 + 2g_2 + 2g_3 + g_4)/6 \tag{8.37}$$

cu

$$\begin{aligned} g_1 &= f(x_k, t_k); \\ g_2 &= f\left(x_k + \tfrac{h}{2} g_1, t_k + \tfrac{h}{2}\right); \\ g_3 &= f\left(x_k + \tfrac{h}{2} g_2, t_k + \tfrac{h}{2}\right); \\ g_4 &= f(x_k + h g_3, t_k + h). \end{aligned}$$

Această metodă calculează panta g_1, în punctul iniţial (x_1, t_1), pe care o foloseşte pentru a avansa cu jumătate de pas $h/2$, în vederea calculului pantei g_2. Valoarea g_2 astfel calculată este folosită pentru a recalcula panta g_3 la jumătatea pasului. Acestă pantă este folosită pentru calculul valorii $f(x,t)$ a funcţiei g_4 la sfârşitul pasului de timp. Cele patru valori obţinute sunt apoi mediate cu ponderile 1/6, 2/6, 2/6 şi respectiv 1/6, în vederea calculului noii stări x_{k+1}.

Această metodă are avantajul unei erori de trunchiere relativ mici, de tipul $O(h^4)$ şi a faptului că permite o implementare simplă în limbajele de programare. Efortul constă, în principal, în evaluarea funcţiei $f(x,t)$ de patru ori la fiecare pas de timp.

Folosirea metodelor de tip Runge-Kutta de **ordin mai mare de 4** nu este justificată, deoarece pentru obţinerea unei erori de tip $O(h^5)$ este necesară evaluarea funcţiei $f(x,t)$

de şase ori. Metodele de ordin 6, 7 şi 8 necesită un număr de 7, 9 şi respectiv 11 evaluări ale funcţiei $f(x,t)$ [?].

Pentru studiul **stabilităţii numerice absolute** se consideră din nou ecuaţia test (8.11), la care $f(x,t) = \lambda x$. Metoda Runge-Kutta de ordinul doi, aplicată acestei ecuaţii, conduce la:

$$x_{k+1} = x_k + h[a_1\lambda x_k + a_2\lambda(x_k + b_1 h\lambda x_k)] = x_k\left[1 + \lambda h + \frac{(\lambda h)^2}{2}\right], \tag{8.38}$$

şi respectiv pentru soluţia perturbată:

$$z_{k+1} = z_k\left[1 + \lambda h + \frac{(\lambda h)^2}{2}\right].$$

Perturbaţia de la pasul k, $\delta_k = z_k - x_k$, se propagă la pasul $k+1$, sub forma:

$$\delta_{k+1} = z_{k+1} - x_{k+1} = \left[1 + \lambda h + \frac{(\lambda h)^2}{2}\right]\delta_k.$$

Condiţia ca această perturbaţie să se atenueze de la un pas la altul este:

$$\left[1 + \lambda h + \frac{(\lambda h)^2}{2}\right] \leq 1, \tag{8.39}$$

ceea ce corespunde în planul complex λh unei regiuni de stabilitate absolută de forma celei reprezentate în figura 8.11a. Dacă se analizează în mod asemănător metoda Runge-Kutta de ordinul patru, rezultă condiţia:

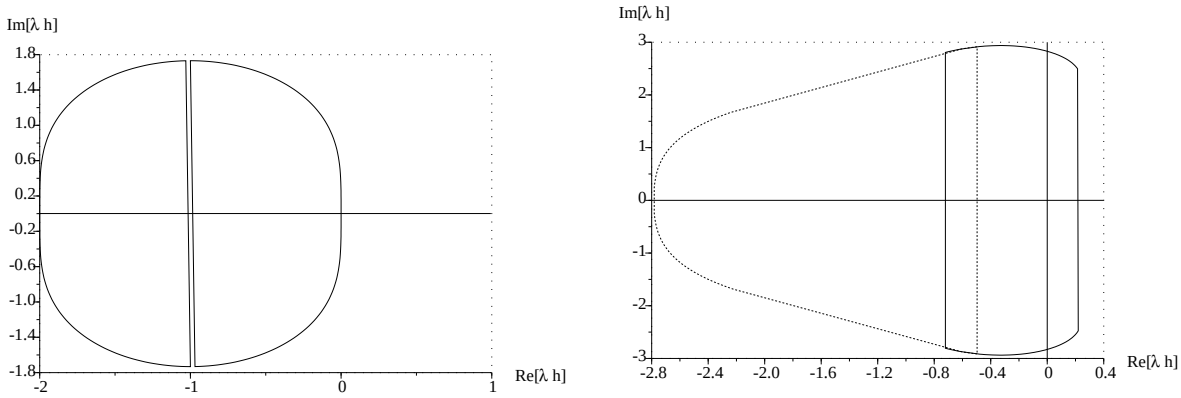
$$\left[1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24}\right] \leq 1, \tag{8.40}$$

care este reprezentată grafic în figura 8.11b.

Prin compararea regiunilor de stabilitate numerică absolută, rezultă că metodele de ordin superior admit, pentru aceeaşi ecuaţie, valori mai mari ale pasului maxim admisibil h, decât metodele de ordin inferior.

Metoda Runge-Kutta de ordinul patru prezintă avantaje faţă de metoda Euler. De exemplu, pentru un pas de integrare $h = 0.1\tau$, metoda Runge-Kutta asigură o eroare cu trei ordine de mărime mai mică decât metoda Euler cu acelaşi pas. Pentru a obţine o eroare comparabilă, pasul în metoda Euler trebuie micşorat la $h = 10^{-4}\tau$, ceea ce nu numai că sporeşte efortul de calcul de câteva sute de ori, dar şi determină o creştere inadmisibilă a erorilor de rotunjire. Această afirmaţie se dovedeşte a fi adevărată în cazul în care se urmăresc erori mici de trunchiere. Dacă toleranţa erorii este mare, metoda Euler poate fi mai avantajoasă din punctul de vedere al efortului de calcul.

Următorul algoritm rezolvă un sistem de N ecuaţii diferenţiale de ordinul unu prin metoda Runge-Kutta de rang 4.



a) b)

Figura 8.11: Domeniile de stabilitate absolută pentru metodele Runge-Kutta de ordinul 2 (a) şi 4 (b).

```
procedura runge_4 (t0,tf,h,y)   ;Rezolvă sistem de N ecuaţii
                                ;diferenţiale de ordin I,
                                ;prin metoda Runge-Kutta de
                                ;rang 4.
real t0,          ;moment iniţial
     tf,          ;limită interval de integrare
     h            ;pas de integrare
tablou real y[N], s[N], z[N], g[N]       ;N-dimensiune
                                         ;sistem
t = t0
cât timp t < tf
        pentru i = 1,N
                g(i) = func (i,y(i),t)           ;evaluare g1
        t = t + h/2
        pentru i = 1,N
                z(i) = y(i) + h·g(i)/2           ;argument pentru g2
                s(i) = g(i)                      ;salvează g1
        pentru i = 1,N
                g(i) = func (i,z(i),t)           ;evaluare g2
        pentru i = 1,N
                z(i) = y(i) + h·g(i)/2           ;argument pentru g3
                s(i) = s(i) + 2·g(i)             ;salvez g1+2·g2
        pentru i = 1,N
                g(i) = func (i,z(i),t)           ;evaluare g3
        t = t + h/2
        pentru i = 1,N
                z(i) = y(i) + h·g(i)             ;argument pentru g4
                s(i) = s(i) + 2·g(i)             ;salvez
```

```
                                                    ;g1+2·g2+2·g3
        pentru i = 1,N
                g(i) = func (i,z(i),t)              ;evaluare g4
        pentru i = 1,N
                y(i) = y(i) + h·(s(i) + g(i))/6     ;soluţie finală
                scrie y(i)                          ;salvează soluţie
retur
```

În figurile 8.12 – 8.14 sunt prezentate soluţia exactă şi soluţiile numerice prin metoda Runge-Kutta de ordinul patru. Se constată că metoda este robustă, în mod asemănător cu metoda Euler implicită (vezi figurile 8.6 – 8.8). Totuşi, metoda Runge-Kutta asigură erori mult mai mici decât metoda Euler explicită, aşa cum rezultă din tabelul 8.3.2, în care sunt prezentate eroarea locală la primul pas şi eroarea globală, la rezolvarea ecuaţiei diferenţiale

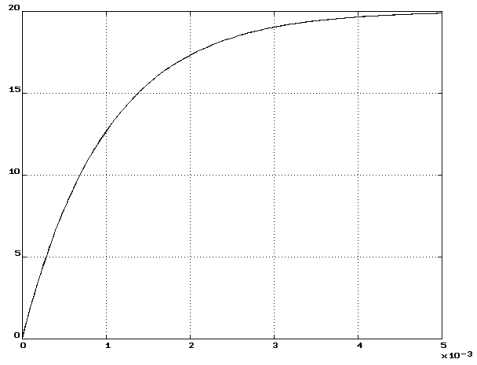


Figura 8.12: Tensiunea la bornele condensatorului pentru $h = \tau/10$

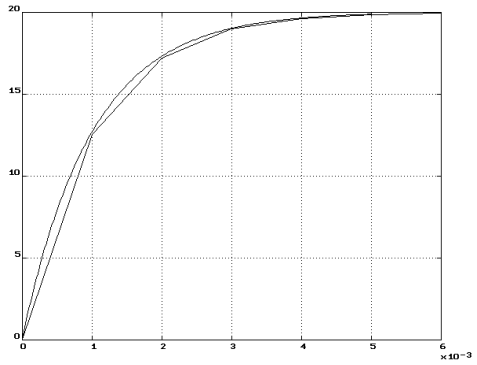


Figura 8.13: Tensiunea la bornele condensatorului pentru $h = \tau$

$$\frac{\partial y}{\partial t} = -y,$$
$$y(0) = 1,$$

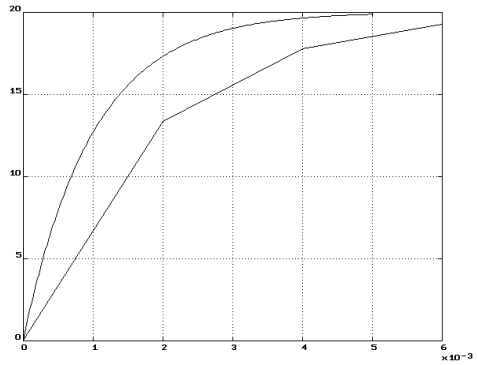


Figura 8.14: Tensiunea la bornele condensatorului pentru $h = 2\tau$

Tabela 8.1: Erori la rezolvarea ecuaţiei diferenţiale $dy/dt = -y, \quad y_0 = 1 \quad h \in [0, 4]$

	Euler	explicit	R–K	ord.4
h	e_l	e_g	e_l	e_g
0.001	$4.9 \cdot 10^{-7}$	$-9.8 \cdot 10^{-1}$	0	$5.3 \cdot 10^{-15}$
0.01	$4.9 \cdot 10^{-5}$	$-9.8 \cdot 10^{-1}$	$-8.3 \cdot 10^{-13}$	$-6.2 \cdot 10^{-12}$
0.1	$4.8 \cdot 10^{-3}$	$-9.8 \cdot 10^{-1}$	$-8.2 \cdot 10^{-8}$	$-6.6 \cdot 10^{-8}$
1	$3.7 \cdot 10^{-1}$	$-9.8 \cdot 10^{-1}$	$-7.1 \cdot 10^{-3}$	$-1.5 \cdot 10^{-3}$
2	1.13	$-9.8 \cdot 10^{-1}$	$-1.9 \cdot 10^{-1}$	$-9.3 \cdot 10^{-2}$

pe intervalul $[0, 4]$.

Graficele la rezolvarea cu metoda Runge-Kutta de ordinul patru a sistemelor de ecuaţii corespunzătoare circuitului din figura 8.15 sunt prezentate în figurile 8.16 – 8.18.

Circuitul din figura 8.15 este caracterizat de sistemul de ecuaţii:

$$\begin{cases} Li' + u + R_i = E, \\ Cu' = i, \end{cases}$$

sistem care poate fi adus la forma:

$$\begin{bmatrix} u' \\ i' \end{bmatrix} = \begin{bmatrix} o & \frac{1}{C} \\ -\frac{1}{L} & -\frac{\hat{R}}{L} \end{bmatrix} \begin{bmatrix} u \\ i \end{bmatrix} + \begin{bmatrix} E \\ 0 \end{bmatrix}.$$

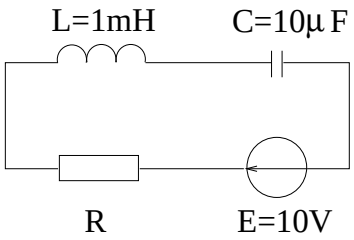


Figura 8.15: Circuit de ordinul II

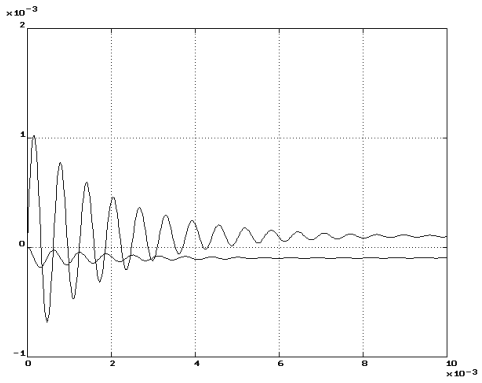


Figura 8.16: Soluţia circuitului din figura 8.15 pentru $R = 1\Omega$

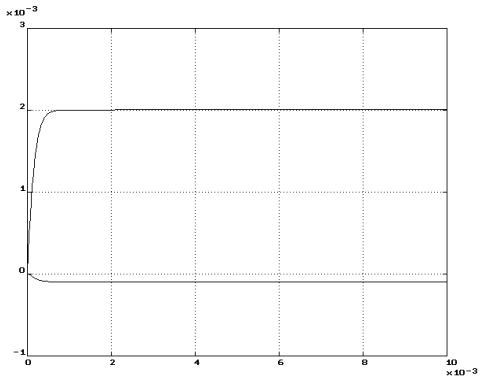


Figura 8.17: Soluţia circuitului din figura 8.15 pentru $R = 20\Omega$

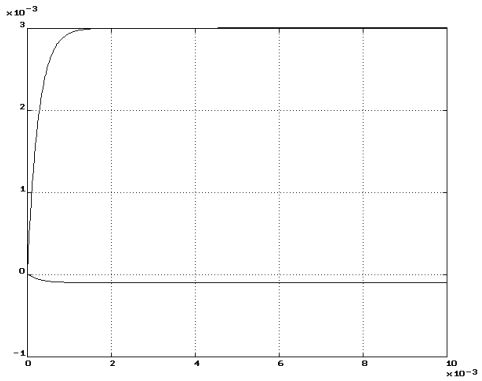


Figura 8.18: Soluţia circuitului din figura 8.15 pentru $R = 30\Omega$

8.3.3 Controlul automat al mărimii pasului de integrare

Criteriile prezentate privind alegerea pasului de integrare pot fi aplicate în analiza circuitelor electrice simple. În cazul circuitelor complexe, parametrice sau neliniare, constantele de timp nu sunt cunoscute aprioric sau nu pot fi definite. În aceste cazuri prezintă interes tehnicile de **alegere automată a pasului** în timpul integrării. De obicei se urmăreşte creşterea pasului de integrare până la limita la care erorile de trunchiere sunt încă admisibile.

O metodă relativ simplă pentru alegerea pasului şi pentru controlul erorii de integrare constă în dublarea (sau înjumătăţirea) pasului de timp h, astfel încât eroarea de integrare să fie cuprinsă permanent între două limite impuse $\varepsilon_m < \varepsilon < \varepsilon_M$. Pentru estimarea erorii de integrare ε se aplică relaţiile (8.37) pentru doi paşi succesivi:

$$\begin{aligned} \widetilde{\mathbf{x}}_{k+1} &= R(\mathbf{x}_k, t_k, h), \\ \widetilde{\widetilde{\mathbf{x}}}_{k+2} &= R(\mathbf{x}_{k+1}, t_{k+1}, h), \end{aligned}$$

şi apoi pentru un pas de lungime dublă:

$$\widetilde{\widetilde{\mathbf{x}}}_{k+2} = R(\mathbf{x}_k, t_k, 2h).$$

Dacă erorile de integrare sunt neglijabile, valorile $\widetilde{x}_{k+2}$ şi $\widetilde{\widetilde{x}}_{k+2}$ astfel calculate trebuie să fie egale. Diferenţa dintre aceste valori poate fi considerată ca o măsură a erorii de integrare:

$$\varepsilon = ||\widetilde{\mathbf{x}}_{k+2} - \widetilde{\widetilde{\mathbf{x}}}_{k+2}||. \tag{8.41}$$

Dacă $\varepsilon > \varepsilon_M$ atunci pasul de timp h se înjumătăţeşte iar dacă $\varepsilon < \varepsilon_m$ atunci pasul de timp se dublează.

Această metodă necesită un efort de calcul sporit, la fiecare pas de timp fiind necesare în medie 5.5 evaluări ale funcţiei $f(x,t)$. În implementarea algoritmului trebuie acordată o atenţie deosebită modului în care se calculează norma erorii. Fiecare componentă trebuie ponderată în mod corespunzător, de exemplu dacă o componentă reprezintă curentul printr-o bobină şi alta sarcina unui condensator, atunci acestea trebuie raportate la valoarea tipică a curentului din bobină şi respectiv la sarcina tipică din condensator. În acest fel eroarea obţinută are un caracter relativ. Eroarea maximă ε_M recomandată este de circa 10^{-3} iar eroarea minimă ε_m nu trebuie să fie mai mică decât eroarea de rotunjire (se recomandă valori $\varepsilon_m = 10^{-5}$, dacă se lucrează în simplă precizie).

O îmbunătăţire a erorii de calcul se poate obţine dacă se adoptă pentru soluţia numerică valoarea calculată prin extrapolare Richardson, pornind de la $\widetilde{\mathbf{x}}_{k+2} = \mathbf{r}(t_{k+2}, h)$ şi $\widetilde{\widetilde{\mathbf{x}}}_{k+2} = \mathbf{r}(t_{k+2}, 2h)$. Extrapolând funcţia $\mathbf{r}(t_{k+2}, y)$ pentru $y = 0$ se obţine:

$$\mathbf{x}_{k+2} = r(t_{k+2}, 0) = 2\widetilde{x}_{k+2} - \widetilde{\widetilde{x}}_{k+2}.$$

Procedeul de extrapolare Richardson poate fi generalizat cu bune rezultate la ordine mai mari, folosind simultan mai multe reţele de discretizare cu paşii h, $h/2$, $h/4$, $h/6$, $h/8$...

[26]. În acest scop, se consideră că soluţia numerică admite, pentru valori mici ale mărimii pasului h, o aproximare polinomială:

$$\mathbf{x}_k^{(h)} = \mathbf{r}(t_k, h) = \mathbf{y}_0 + \mathbf{y}_1 h + \mathbf{y}_2 h^2 + ...$$

Este de aşteptat ca valoarea acestei aproximări pentru $h = 0$ să fie mai aproape de soluţia exactă decât pentru $h \neq 0$. Coeficienţii extrapolării, din care interesează mai ales y_0, se determină cu relaţiile Aitken- Lagrange folosind reţele din ce în ce mai fine, până când abaterea dintre două soluţii scade sub toleranţa impusă.

Utilizând extrapolarea raţională în locul celei polinomiale, Bulirsch şi Stoer au obţinut una din cele mai eficiente metode de mare acurateţe pentru rezolvarea numerică a ecuaţiilor diferenţiale [2], [26].

Pentru a asigura pemanent o valoare optimă a pasului de integrare Gear [26] propune o metodă bazată pe o singură eroare impusă ε_M. Valoarea pasului de timp h_k se modifică la fiecare etapă de integrare conform relaţiei

$$h_{k+1} = 0,99 h_k \left(\frac{15\varepsilon_M ||x_k||}{\varepsilon} \right)^{\frac{1}{5}},$$

care asigură o eroare de trunchiere cât mai apropiată de ε_M. Dacă $\varepsilon/||x_k||$ depăşeşte toleranţa $15\varepsilon_M$ atunci calculul se reia cu noul pas de timp. Valoarea adoptată pentru soluţia numerică este o medie ponderată a soluţiilor obţinute cu cei doi paşi de timp:

$$\mathbf{x}_{k+2} = (16\widetilde{\mathbf{x}}_{k+2} - \widetilde{\widetilde{\mathbf{x}}}_{k+2})/15.$$

O metodă ingenioasă de modificare a pasului este propusă de Zonnenveld [2]. Acesta adoptă ca măsură a erorii norma ultimului termen din seria Taylor trunchiată, respectiv:

$$q_k = \left|\left| \frac{h_k^p}{p!} \mathbf{x}_k^{(p)} \right|\right|. \tag{8.42}$$

Se urmăreşte calculul acestui termen în funcţie de evaluările deja efectuate ale funcţiei $\mathbf{f}(\mathbf{x}, t)$. În cazul metodei Runge-Kutta de ordinul doi (8.29) se determină coeficienţii d_1 şi d_2 astfel încât:

$$\frac{h^2}{2!} \mathbf{x}_k^{(2)} = h[d_1 \mathbf{g}_1 + d_2 \mathbf{g}_2]. \tag{8.43}$$

Utilizând relaţiile (8.30), (8.31) rezultă:

$$\frac{h^2}{2} \left[\frac{\partial \mathbf{f}}{\partial x} g_1 + \frac{\partial \mathbf{f}}{\partial t} \right] = h \mathbf{g}_1 (d_1 + d_2) + h^2 \frac{\partial f}{\partial x} \mathbf{g}_1 b_1 d_2 + \frac{\partial f}{\partial t} b_2 d_2,$$

ceea ce impune constantelor d_1 şi d_2 relaţiile:

$$d_1 + d_2 = 0 \ ; \ d_2 b_1 = 1/2 \ ; \ d_2 b_2 = 1/2. \tag{8.44}$$

Alegând, de exemplu, varianta Heun a metodei Runge-Kutta, rezultă $d_1 = -1/2,\ d_2 = 1/2,$ deci eroarea de la pasul k poate fi măsurată prin:

$$q_k = \frac{h_k}{2} \left|\left| \mathbf{f}(\mathbf{x}_k, t_k) - \mathbf{f}(\mathbf{x}_k + h\mathbf{f}(\mathbf{x}_k, t_k), t_k + h) \right|\right| . \tag{8.45}$$

Folosind o tehnică asemănătoare se obţine norma ultimului termen şi în cazul metodei Runge-Kutta de ordinul patru (8.37):

$$q_k = \left|\left| \frac{h_k^4}{4!} x_k^{(4)} \right|\right| = h_k \left|\left| -\frac{2}{3} g_1 + 2g_2 + 2g_3 + 2g_4 - \frac{16}{3} g_5 \right|\right| , \tag{8.46}$$

cu

$$q_5 = f\left(x_k + \frac{h}{32}(5g_1 + 7g_2 + 13g_3 - g_4), t_k + \frac{3}{4}h \right) .$$

Se constată că în cazul metodei de ordinul patru este necesară o evaluare suplimentară a funcţiei $\mathbf{f}(\mathbf{x}, t)$. Eroarea q_k de la pasul curent este comparată cu:

$$\tau_k = h_k[\varepsilon_1 ||f(x_k, t_k)|| + \varepsilon_2], \tag{8.47}$$

unde ε_1 şi ε_2 sunt două valori limită date, impuse erorii.

Dacă $q_k \leq \tau_k$ atunci valoarea (x_{k+1}) calculată la pasul curent este acceptată ca fiind corectă şi se trece la pasul următor h_{k+1}, care se majorează faţă de pasul curent. Dacă eroarea q_k depăşeşte toleranţa admisă $(q_k > \tau_k)$, atunci pasul curent h_k se micşorează şi se recalculează valoarea x_{k+1} cu noul pas.

Pentru coeficientul de corecţie al pasului, Zonnenveld propune valoarea:

$$\mu_k = \frac{\tau_k}{\tau_k + q_k} + 0.45. \tag{8.48}$$

Dacă rezultatul este eronat $(q_k < \tau_k)$ atunci pasul este corectat la valoarea $\mu_k h_k$ (cu $0.45 < \mu_k < 0.95$). Dacă rezultatul este acceptat, atunci coeficientul de corecţie satisface inegalitaţile $0.95 \leq \mu_k \leq 1.45$. Pentru a evita o mărire hazardată a pasului, Zonnenveld propune strategia dată de relaţia:

$$h_{k+1} = \begin{cases} \mu_k h_k \ , & pentru\ k = 0; \\ h_k \left[\frac{h_k}{h_{k-1}} (1 + \mu_k) - \mu_{k-1} \right] \ , & pentru\ k > 0. \end{cases} \tag{8.49}$$

O variantă a metodei Runge-Kutta de ordinul patru, optimizată din punctul de vedere al efortului de calcul (şi al numărului de registre procesor utilizate) se datorează lui Gill, care alege în acest sens în (8.36), $b_6 = 1 + 1/\sqrt{2}$ şi $c_1 = 1/2$. În varianta sa finală,

algoritmul lui Gill are forma [48], [26]:

$$\begin{aligned}
g_1 &= hf(x_k, t_k),\\
&\quad q_1 = q_0 + 3(g_1 - 2q_0)/2 - g_1/2,\\
&\quad y_1 = x_k + (g_1 - 2q_0)/2,\\
g_2 &= hf(y_1, t_k + h/2),\\
&\quad q_2 = q_1 + 3(1 - 1/\sqrt{2})(q_2 - q_1) - (1 - 1/\sqrt{2})q_2,\\
&\quad y_2 = y_1 + (1 - 1/\sqrt{2})(q_2 - q_1),\\
g_3 &= hf(y_2, t_k + h/2),\\
&\quad q_3 = q_2 + 3(1 + 1/\sqrt{2})(q_3 - q_2) - (1 + 1/\sqrt{2})q_3,\\
&\quad y_3 = y_2 + (1 + 1/\sqrt{2})(q_3 - q_2),\\
g_4 &= hf(y_3, t + h),\\
&\quad q_4 = q_3 + 3(q_4 - 2q_3)/6 - q_4/2,\\
x_{k+1} &= y_4 = y_3 + (q_4 - 2q_3)/6.
\end{aligned} \tag{8.50}$$

El asigură un număr minim de operaţii şi folosirea cu eficienţă maximă a memoriei. Mai mult, această metodă permite compensarea parţială a erorii de rotunjire, de la un pas la altul. În acest scop se alege iniţial $q_0 = 0$ iar dacă precizia ar fi infinită ar trebui să rezulte $q_4 = 0$. În realitate acest lucru nu este adevărat şi q_4 reprezintă circa triplul erorii de rotunjire de care este afectat rezultatul y_4. Pentru a compensa această eroare, în pasul următor se iniţializează q_0 cu valoarea q_4 de la pasul precedent.
În implementarea algoritmului Runge-Kutta-Gill, dată în [48], controlul erorii şi al pasului se realizează cu ajutorul relaţiei (8.41), limita inferioară a erorii fiind aleasă $\varepsilon_m = \varepsilon_M/50$.

8.4 Metode de integrare multipas

Metodele Taylor şi Runge-Kutta fac parte din categoria metodelor de integrare a ecuaţiilor diferenţiale cu un pas, deoarece permit estimarea soluţiei x_k, la pasul de timp k, folosind doar valoarea x_{k-1} de la pasul anterior. Trebuie observat că pe parcursul integrării, sunt deja cunoscute valorile soluţiei într-o serie de paşi anteriori $x_{k-1}, x_{k-2}, \ldots$, valori care la metodele cu un pas nu sunt luate în considerare. Utilizarea acestor valori în calculul soluţiei de la pasul curent poate duce la creşterea preciziei fără un efort de calcul suplimentar. Metodele care în determinarea soluţiei folosesc informaţiile de la mai mulţi paşi, anteriori pasului curent se numesc *metode multipas*. Aceste metode nu au autostart, deoarece la primul pas este cunoscută doar condiţia iniţială x_0. Din acest motiv, integrarea trebuie demarată cu o metodă cu un pas, de regulă cu Runge-Kutta care este utilizată la integrarea ecuaţiei pe un număr de paşi, şi apoi se continuă integrarea folosind metode multipas.

Se consideră o reţea uniformă de discretizare a variabilei independente:

$$t_0,\ t_1 = t_0 + h,\ t_2 = t_o + 2h,\ \ldots,\ t_k = t_0 + k_h, \ldots$$

şi se presupun cunoscute valorile soluţiei numerice în primele noduri ale reţelei:

$$x_0,\ x_1,\ x_2,\ \ldots,\ x_{k-1},$$

precum şi valorile derivatelor

$$x'_0 = f(x_0, t_0),\ x'_1 = f(x_1, t_1),\ \ldots,\ x'_{k-1} = f(x_{k-1}, t_{k-1}).$$

Pentru obtinerea relaţiei de recurenţă specifică metodelor implicite se integrează derivata soluţiei pe un număr de n paşi succesivi:

$$\hat{x}(t_k) - \hat{x}(t_{k-n}) = \int_{t_{k-n}}^{t_k} \hat{x}'(t)dt. \tag{8.51}$$

Folosind o metodă de cuadratură numerică, integrala va fi aproximată printr-o combinaţie liniară de valori ale integrandului de pe ultimile p noduri ale reţelei de discretizare:

$$x_k - x_{k-n} = h \sum_{i=1}^{p} b_i x'_{k-i},$$

ceea ce generează relaţia de recurenţă:

$$x_k = x_{k-n} + h[b_1 f(x_{k-1}, t_{k-1}) + \ldots + b_p f(x_{k-p}, t_{k-p})]. \tag{8.52}$$

Relaţia (8.52) poate fi generalizată sub forma:

$$x_k = \sum_{i=1}^{p} (a_i x_{k-i} + b_i h x'_{k-i}), \tag{8.53}$$

în care trebuie determinaţi coeficienţii $a_1, a_2, \ldots, b_1, b_2, \ldots$. Aceşti coeficienţi, în număr de $2p$, ponderează valorile soluţiei şi respectiv ale derivatei acesteia pe ultimii p paşi de integrare. Relaţia (8.53) pune în evidenţă caracterul **explicit** al acestei clase de metode. Pentru metodele multipas, parametrul p din (8.53) indică numărul de paşi ai metodei, dar în particular dacă se alege $p = 1$ se obţine pentru $a_1 = b_1 = 1$ metoda cu un pas, de tip Euler explicit.

Dacă în aproximarea integralei (8.51) se foloseşte inclusiv valoarea integrandului din punctul final t_k, se obţine:

$$x_k - x_{k-n} = h \sum_{i=0}^{p} b_i x'_{k-i},$$

relaţie care poate fi scrisă în general, sub forma implicită:

$$\sum_{i=0}^{p} (a_i x_{k-i} + b_i h x'_{k-i}) = 0. \tag{8.54}$$

Relaţia (8.54) generalizează relaţia (8.53), care se poate obţine prin alegerea $a_0 = -1, b_0 = 0$. Spre deosebire de (8.53), relaţia (8.54) conţine un singur termen suplimentar $x'_k = f(x_k, t_k)$.

Dacă se normează (8.54), astfel încât $a_0 = -1$, rezultă

$$x_k = b_0 h f(x_k, t_k) + \sum_{i=1}^{p} [a_i x_{k-i} + b_i h f(x_{k-i}, t_{k-i})]. \tag{8.55}$$

Se arată că (8.55) evidenţiază caracterul implicit al clasei de metode obţinute. Soluţia numerică x_k de la pasul k intervine atât în primul membru cât şi sub funcţia $f(x,t)$ din membrul doi. Dacă $f(x,t)$ este o funcţie neliniară, atunci determinarea soluţiei x_k presupune rezolvarea unui sistem de ecuaţii neliniare (8.55). Aplicarea metodelor implicite necesită deci un efort de calcul sporit faţă de cele explicite.

8.4.1 Metoda explicită Adams-Bashforth

Metoda Adams-Bashforth se bazează pe relaţia de recurenţă explicită (8.52), în care se alege $n = 1$, ceea ce corespunde alegerii $a_1 = 1, a_2 = 0, a_3 = 0 \ldots$ în relaţia (8.53). Coeficienţii b_i se pot determina prin calculul integralei (8.51) cu metoda de extrapolare polinomială de tip Newton, bazată pe diferenţe regresive. În continuare, aceşti coeficienţi vor fi determinaţi, pentru cazul $p = 2$ (cu doi paşi), totuşi cu metoda seriei Taylor trunchiate, pentru a evidenţia şi ordinul erorii de trunchiere. Alegând în (8.53) $p = 2$ şi $a_2 = 0$, rezultă:

$$x_k = a_i x_{k-1} + h b_1 x'_{k-1} + h b_2 x'_{k-2}, \tag{8.56}$$

iar pe baza dezvoltărilor Taylor:

$$\begin{aligned} \hat{x}(t_k) \;=\; & a_1 \left[\hat{x}(t_k) - h\hat{x}'(t_k) + \frac{h^2}{2}\hat{x}'(t_k) + O(h^3) \right] + \\ & + h b_1 \left[\hat{x}'(t_k) - h\hat{x}'(t_k) + O(h^2) \right] + \\ & + h b_2 \left[\hat{x}'(t_k) - 2h\hat{x}'(t_k) + O(h^3) \right]. \end{aligned} \tag{8.57}$$

Identificând termenii în h_0, h_1, h_2, rezultă sistemul:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \\ \frac{1}{2} & -1 & -2 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ b_2 \end{bmatrix}, \tag{8.58}$$

care are soluţia $a_1 = 1, b_1 = \frac{3}{2}, b_2 = \frac{-1}{2}$, ceea ce corespunde relaţiei de integrare numerică Adams-Bashforth cu doi paşi:

Tabela 8.2: Coeficienţii metodei Adams-Bashforth

p	i	1	2	3	4	5	6
1	b_i	1					
2	$2b_i$	3	-1				
3	$12b_i$	23	-16	5			
4	$24b_i$	55	-59	37	-9		
5	$720b_i$	1901	-2774	2616	-1274	251	
6	$1440b_i$	4277	-7293	9982	-7298	2877	-475

$$x_{k+1} = x_k + \frac{h}{2}[3x'_k - x'_{k-1}]. \tag{8.59}$$

Relaţia (8.57) evidenţiază eroarea locală de trunchiere a metodei cu doi paşi, care este de tipul $O(h^3)$. Relaţia (8.56) trebuie să fie satisfăcută identic de un polinom de gradul doi, deci de funcţiile $x = 1, s, s^2$ cu $s = (t - t_k)/h$. Aceasta este echivalentă cu condiţia ca primii trei termeni ai seriei Taylor să se anuleze, deci condiţiile de interpolare polinomială vor conduce la aceleaşi valori (8.59) ale coeficienţilor.

În general se consideră funcţia polinomială:

$$\hat{x}(t) = s^n, \hat{x}'(t) = \frac{ns^{n-1}}{h},$$

unde $s = (t - t_{k-1})/h$ si $n = 1, 2, \ldots p$, care se substituie în $x_k = x_{k-1} + h\sum_{i=1}^{p} b_i x'_{k-i}$ şi se obţine:

$$1 = n\sum_{i=1}^{p} b_i s^{n-1}\Big|_{t=t_k-h} = n\sum_{i=1}^{p}(1-i)^{n-1}b_i\ . \tag{8.60}$$

Coeficienţii b_i satisfac deci sistemul de ecuaţii:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 0 & -1 & -2 & -3 & \ldots & -(p-1) \\ 0 & 1 & 4 & 9 & \ldots & (p-1)^2 \\ 0 & -1 & -8 & -27 & \ldots & -(p-1)^3 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & (-1)^{p-1} & (-2)^{p-1} & (-3)^{p-1} & \ldots & (1-p)^{p-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \ldots \\ b_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/4 \\ \ldots \\ 1/p \end{bmatrix}, \tag{8.61}$$

care pentru diferite ordine are soluţiile din tabelul 8.2:

Se poate afirma că la metoda Adams-Bashforth cu p paşi eroarea locală de trunchiere satisface inegalitatea:

$$\varepsilon = ||x(t_k) - x_k|| \leq C_p h^{p+1}, \tag{8.62}$$

deci $\varepsilon = O(h^{p+1})$ [9],[4], având ordinul $p+1$, urmând ca eroarea globală să aibă ordinul egal cu numărul de paşi.

Se constată că pentru a obţine o comportare a erorii de trunchiere comparabilă cu cea dată de metoda Runge-Kutta de ordinul patru, trebuie aplicată metoda Adams-Bashforth cu patru paşi:

$$x_{k+1} = x_k + \frac{h}{24}[55f(x_k, t_k) - 59x'_{k-1} + 37x'_{k-2} - 9x'_{k-3}], \tag{8.63}$$

ceea ce presupune o singură evaluare a funcţiei $f(x,t)$ în punctul (x_k, t_k), urmând ca celorlalte trei valori $x'_{k-i} = f(x_{k-i}, t_{k-i}), i = 1,2,3$ să fi fost memorate la paşii anteriori. În consecinţă, efortul de calcul este de circa patru ori mai mic decât la metoda Runge-Kutta. Pentru a putea demara, algoritmul generat de (8.63) are nevoie de o iniţializare, care presupune aplicarea metodei Runge-Kutta pe primii trei paşi: $t_1 = t_0 + h$, $t_2 = t_0 + 2h$, $t_3 = t_0 + 3h$, în vederea calculului valorilor $x'_{k-3} = f(x_0, t_0)$, $x'_{k-2} = f(x_1, t_1)$, $x'_{k-1} = f(x_2, t_2)$ şi $x_k = f(x_3, t_3)$. pentru $k = 3$.

8.4.2 Metoda implicită Adams-Moulton

Metoda Adams-Moulton, fiind implicită, se bazează pe relaţia (8.55), în care ca şi în cazul anterior se alege $a_1 = 1$ şi $a_2 = a_3 = \ldots = a_p = 0$, ceea ce corespunde alegerii $n = 1$, specifice metodelor de tip Adams:

$$x_k = x_{k-1} + h\sum_{i=1}^{p} b_i x'_{k-i}. \tag{8.64}$$

Dacă relaţia (8.64) este satisfacută identic de un polinom arbitrar de gradul $(p+1)$, atunci eroarea locală de trunchiere are un ordin cu o unitate mai mare decât gradul polinomului $O(h^{p+2})$. După cum s-a arătat în paragraful anterior, această condiţie este echivalentă cu anularea primilor $(p+2)$ termeni din seria Taylor asociată soluţiei exacte. Dacă se substituie în (8.64) polinoamele:

$$\hat{x}(t) = s^n, \text{ cu } n = 1, 2, \ldots, (p+1),$$

se obţine sistemul de ecuaţii satisfăcut de b_i:

$$\sum_{i=0}^{p}(1-i)^{n-1}b_i = 1/n, n = 1, 2, \ldots, (p+1)$$

sau dezvoltat:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 1 & 0 & -1 & -2 & \ldots & -(p-1) \\ 1 & 0 & 1 & 4 & \ldots & (1-p)^2 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & (-1)^{p-1} & (-2)^{p-1} & \ldots & (1-p)^{p-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ \vdots \\ 1/(p+1) \end{bmatrix} \tag{8.65}$$

Tabela 8.3: Coeficienţii metodei Adams-Moulton

p	i=	0	1	2	3	4	5
0	b_i	1					
1	$2b_i$	1	1				
2	$12b_i$	5	8	-1			
3	$24b_i$	9	19	-5	1		
4	$720b_i$	251	646	-264	106	-19	
5	$1440b_i$	475	1427	-798	482	-173	27

Particularizând sistemul (8.65) pentru cazul $p = 1$:

$$\begin{cases} b_0 + b_1 = 1; \\ b_0 = 1/2; \end{cases}$$

se obţin valorile coeficienţilor $b_0 = b_1 = 1/2$. Relaţia de integrare Adams-Moulton cu un pas:

$$x_{k+1} = x_k + \frac{h}{2}[f(x_k, t_k) + f(x_{k+1}, t_{k+1})] \tag{8.66}$$

este echivalentă cu metoda trapezelor aplicată integralei (8.51). Această variantă a metodei asigură o eroare locală de trunchiere de tipul $O(h^3)$, deci o comportare mai bună a erorii decât la metoda Euler implicită, care se obţine formal pentru $p = 0$ şi care necesită practic acelaşi efort de calcul. Metoda trapezelor fiind cu un singur pas are autostartul asigurat, dar necesită rezolvarea la fiecare pas de timp a ecuaţiei (8.66), în vederea determinării soluţiei x_{k+1}. Dacă se utilizează rezolvarea iterativă, atunci se recomandă ca iniţializarea să se facă cu valoarea dată de metoda Euler explicită. Se constată în acest caz că prima iteraţie are valoarea corespunzătoare metodei Runge-Kutta de ordinul doi în varianta Heun.

În cazul metodei implicite cu doi paşi ($p = 2$) se obţine $b_0 = 5/12, b_1 = 8/12, b_2 = -1/12$, ceea ce corespunde la:

$$x_{k+1} = x_k + \frac{h}{12}[5f(x_{k+1}, t_{k+1}) + 8f(x_k, t_k) - f(x_{k-1}, t_{k-1})], \tag{8.67}$$

relaţie pe doi paşi care asigură o eroare locală de trunchiere de tipul $O(h^4)$. Tabelul 8.3 conţine coeficienţii metodei Adams-Moulton până la variantele cu cinci paşi.

În metoda Adams-Moulton eroarea locală de trunchiere satisface inegalitatea:

$$\varepsilon = ||\hat{x}(t_k) - x_k|| \leq C_p h^{p+2}, \tag{8.68}$$

deci ordinul metodei implicite este cu o unitate mai mare decât cel al metodei explicite cu acelaşi număr de paşi. Pentru a obţine o comportare a erorii de trunchiere de acelaşi tip cu metoda Runge-Kutta de ordinul patru este suficientă aplicarea metodei implicite cu trei paşi:

$$x_{k+1} = x_k + \frac{h}{24}[9f(x_{k+1}, t_{k+1}) + 19x'_k - 5x'_{k-1} + x'_{k-2}]. \tag{8.69}$$

Efortul de calcul în metoda Adams-Moulton este mult mai mare decât cel din metoda Runge-Kutta, deoarece presupune rezolvarea ecuaţiei (8.69) în vederea determinării soluţiei numerice x_{k+1}. Pentru demararea integrării prin relaţia (8.69) este necesară aplicarea pe primii doi paşi de timp $t_0 + h, t_1 + h$ a metodei Runge-Kutta în vederea determinării funcţiilor $x'_{k-2} = fx_0, t_0, x'_{k-1} = f(x_1, t_1)$ şi $x'_k = f(x_2, t_2)$.

O metodă eficientă pentru rezolvarea ecuaţiei implicite constă în calculul iterativ al soluţiei pe baza relaţiei (8.69), pornind de la o anumită iniţializare pentru x_{k+1}. Succesul acestei metode depinde de iniţializare. Dacă pentru iniţializare se foloseşte valoarea dată de o metodă explicită, atunci numărul iteraţiilor necesare se poate reduce la una sau cel mult două. Algoritmul bazat pe folosirea combinată a metodelor explicite şi implicite este cunoscut sub numele **"predictor-corector"**. Din acest motiv metodele implicite se numesc metode de corecţie. Dacă se aplică metoda predictor-corector cu un predictor de tip Adams-Bashforth pe trei paşi:

$$x_{(k+1)p} = x_k + \frac{h}{12}[23f(x_k, t_k) - 16x'_{k-1} + 5x'_{k-2}] \tag{8.70}$$

şi un corector de tip Adams-Moulton pe acelaşi număr de paşi:

$$x_{k+1} = x_k + \frac{h}{24}[9f(x_{(k+1)p}, t_{k+1}) + 19x'_k - 5x'_{k-1} + x'_{k-2}] \tag{8.71}$$

se constată că este necesară evaluarea doar de două ori a funcţiei $f(x, t)$, o dată pentru predictor şi a doua oară pentru corector (sau cel mult de trei ori dacă este necesară o reiterare), faţă de patru ori cât era necesar în metoda Runge-Kutta, pentru asigurarea unei erori comparabile. Mai mult, diferenţa dintre valoarea prezisă şi cea corectată oferă o măsură sigură a erorii de trunchiere:

$$\varepsilon = ||x_{(k+1)p} - x_{k+1}|| \tag{8.72}$$

În consecinţă se poate afirma că metoda predictor-corector este mai avantajoasă din punct de vedere al efortului de calcul şi al controlului erorii decât metoda Runge-Kutta.

Tabela 8.4: Coeficienţii metodei Milne

p	i=	0	1	2	3	4	5
1	b_i	0	2				
3	$3b_i$	0	8	-4	8		
5	$10b_i$	0	33	-42	78	-42	11
2	$3b_i$	1	4	1			
4	$45b_i$	14	64	24	64	14	

8.4.3 Metoda Milne

Metoda Milne uzuală este o metodă implicită cu doi paşi în care se alege $n = 2$, sau echivalent în ecuaţia (8.55), $a_1 = 0, a_2 = 1$:

$$x_k = x_{k-2} + h[b_0 x'_k + b_1 x'_{k-1} + b_2 x'_{k-2}]. \tag{8.73}$$

Pentru determinarea celor trei coeficienţi b_0, b_1, b_2 se impune condiţia ca relaţia (8.73) să fie satisfăcută identic de un polinom arbitrar de gradul trei. Dacă se alege:

$$\hat{x}(t) = s^n$$

cu $s = (t - t_{k-2})/2h$ şi $n = 1, 2, 3$, rezultă prin substituţie în (8.73):

$$\begin{aligned} b_0 + b_1 + b_2 &= 2 \\ b_0 + b_1/2 &= 1 \\ b_0 + b_1/4 &= 2/3 \end{aligned}$$

sistem care admite soluţia $b_0 = 1/3, b_1 = 4/3$, şi $b_2 = 1/3$. În consecinţă metoda Milne se bazează pe relaţia recursivă:

$$x_{k+1} = x_{k-1} + \frac{h}{3}[f(x_{k+1}, t_{k+1}) + 4x'_k + x'_{k-1}] \tag{8.74}$$

şi asigură o comportare a erorii de trunchiere de tipul $O(h^4)$. Metodele de tip Milne au pentru coeficienţii din relaţia (8.55) valorile $a_i = 0$ cu $i \neq p$ şi $a_p = 1$. Tabelul 8.4 conţine valorile coeficienţilor b_i pentru diferite ordine. În cazul metodelor explicite ordinele impare $2p' - 1$ oferă aceeaşi eroare ca cele pare $2p'$, deci sunt preferabile. În cazul implicit ordinele pare $2p'$ oferă aceeaşi eroare ca cele de ordin $2p' + 1$, deci sunt preferate. [7]

Încercarea de a considera $a_2 \neq 0$ în metoda explicită (8.53) determină variante instabile numeric ale metodelor multipas. De exemplu, dacă se presupune $p = 2$ în (8.53) se obţine:

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + b_1 h x'_{k-1} + b_2 h x'_{k-2}. \tag{8.75}$$

Tabela 8.5: Soluţia ecuaţiei de test

pasul k	0	1	2	3	4	5	6
soluţia x_k	0	ε	- 4 ε	21 ε	- 104 ε	521 ε	- 2604 ε

Dacă se impune ca relaţia (8.75) să fie verificată identic de un polinom de gradul trei se obţin următoarele valori ale coeficienţilor:

$$a_1 = -4, \qquad a_2 = 5, \qquad b_1 = 4, \qquad b_2 = 2.$$

şi respectiv o metodă explicită cu doi paşi:

$$x_k = -4x_{k-1} + 5x_{k-2} + h[4x'_{k-1} + 2x'_{k-2}], \tag{8.76}$$

cu o comportare a erorii de tip $O(h^4)$ dar care prezintă puternice instabilităţi numerice. De exemplu, în rezolvarea ecuaţiei diferenţiale

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 0$$

cu condiţia iniţială nulă $x(0) = 0$, a cărei soluţie exactă este $\hat{x}(t) = 0$, pentru o iniţializare $x_1 = \varepsilon$ (o mică eroare de rotunjire), soluţia numerică se amplifică conform tabelului 8.5, în loc să rămână constantă.

8.4.4 Consistenţa, stabilitatea şi convergenţa metodelor multipas

Forma generală a relaţiei de recurenţă specifică metodelor multipas este dată de (8.54). Prin particularizarea acestei relaţii se obţin atât metodele explicite ($b_0 = 0$) cât şi cele implicite ($b_0 \neq 0$). Metodele de tip Adams reprezintă doar o parte din metodele multipas, care pot fi obţinute prin particularizarea relaţiei (8.54). Coeficienţii acestora au fost determinaţi impunând condiţia ca (8.54) să fie satisfăcută identic de polinoame arbitrare. Dacă se aplică această tehnică pentru un polinom de grad n, se pot determina un număr de coeficienţi egal cu numărul coeficienţilor din polinom, deci $n + 1$. Relaţia (8.54) conţine 2 $(p + 1)$ coeficienţi $a_0, a_1, \ldots, a_p, b_0, b_1, \ldots, b_p$ din care unul poate fi ales unitar prin normalizarea ecuaţiei. În consecinţă gradul polinomului folosit ca funcţie test trebuie să satisfacă inegalitatea:

$$2p + 1 \geq n + 1, \tag{8.77}$$

diferenţa dintre cei doi membri reprezentând numărul coeficienţilor care se pot alege arbitrar. Gradul n al polinomului de test impune ordinul erorii de trunchiere, care în forma locală este de tipul $O(h^{n+1})$ iar în forma sa globală este de tipul $O(h^n)$, motiv pentru care el se numeşte şi *ordinul metodei.*

Presupunem pentru început că soluţia ecuaţiei diferenţiale este o constantă $\hat{x}(t) = 1$, ceea ce corespunde alegerii unui polinom de grad $j = 0 : x_{k-i} = 1, x'_{k-i} = 0$. Înlocuind aceste valori în (8.54), rezultă:

$$\sum_{i=0}^{p} a_i = 0, \tag{8.78}$$

care este prima condiţie impusă coeficienţilor metodei generale multipas. Dacă se alege ca polinom de test polinomul de gradul $j = 1$, de exemplu $\hat{x}(t) = t$, atunci prin substituţia valorilor $x_n = t_n$ şi $x'_n = 1$, în (8.54), rezultă pentru $t_{k-i} = (1-i)h$:

$$\sum_{i=0}^{p}(1-i)a_i + \sum_{i=0}^{p} b_i = 0.$$

Pentru gradul doi $\hat{x}(t) = t^2$, $x_n = t_n^2$, $x'_n = 2t_n$ rezultă:

$$\sum_{i=0}^{p}(1-i)^2 a_i + 2\sum_{i=0}^{p}(1-i)b_i = 0,$$

iar în general coeficienţii satisfac relaţiile:

$$\sum_{i=0}^{p}(1-i)^j a_i + j\sum_{i=0}^{p}(1-i)^{j-1} b_i = 0, \tag{8.79}$$

pentru $j = 1, 2, \ldots, n$. Dacă o metodă multipas îndeplineşte condiţiile (8.78) şi (8.79) atunci aceasta garantează că metoda permite calculul exact (în măsura în care erorile de rotunjire sunt nule) cel puţin al soluţiilor care sunt polinoame în t, de grad maxim n. Aceste condiţii sunt necesare pentru a asigura convergenţa metodei.

Definiţia 8.2 *O metodă multipas, definită de relaţia (8.54), ai cărei coeficienţi satisfac relaţiile (8.78), (8.79) se numeşte consistentă.*

Ordinul unei metode consistente este dat de valoarea n a gradului maxim pentru care este îndeplinită relaţia (8.79).

Teorema 8.4 *O metodă de discretizare cu p paşi:*

$$x_k = \sum_{i=0}^{p} a_i x_{n-i} + h\sum_{i=0}^{p} b_i f(x_{n-i}, t_{n-i}),$$

care este consistentă cu ordinul n are o eroare locală de trunchiere:

$$\varepsilon = C_n \hat{x}^{(n+1)}(\tau) h^{n+1} = O(h^{n+1}),$$

cu $-ph < \tau < h$ *şi:*

$$C_n = \frac{1}{(n+1)!}\left[p^{n+1} - \sum_{i=1}^{p-1} a_i(p-i)^{n+1} - (n+1)\sum_{i=1}^{p-1} b_i(p-i)^n\right].$$

Această teoremă este demonstrată în [4]. Metodele de tip Adams-Bashforth sunt consistente şi au ordinul egal cu numărul de paşi pe care se aplică. Metodele Adams-Moulton sunt consistente dar au ordinul cu o unitate mai mare decât numărul de paşi pe care se aplică. Metoda Milne (8.74) şi metoda (8.76) sunt şi ele consistente cu ordinul trei. Consistenţa este o condiţie necesară dar nu suficientă pentru ca o metodă să fie utilă în practică. Consistenţa nu se referă în nici un fel la modul în care se propagă erorile de rotunjire. Metoda generată de relaţiile (8.76) este consistentă dar s-a constatat că erorile de rotunjire sunt amplificate în aşa fel încât după câţiva paşi de integrare acoperă complet soluţia.

Pentru a studia stabilitatea numerică a metodelor multipas se consideră din nou ecuaţia diferenţială simplă:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \lambda x \tag{8.80}$$

căreia i se impune condiţia iniţială $x(0) = 1$, ca ecuaţie de test. Soluţia exactă a acestei ecuaţii este $\hat{x}(t) = e^{\lambda t}$, funcţie care în nodurile unei reţele regulate $t_k = kh$ ia valorile:

$$\hat{x}(t_k) = e^{\lambda t_k} = e^{\lambda kh} = (e^{\lambda h})^k = z_0^k. \tag{8.81}$$

Se constată că soluţia exactă reprezintă o progresie geometrică cu relaţia:

$$z_0 = e^{\lambda h} = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \ldots \tag{8.82}$$

În cazul limită în care $\lambda \to 0$ soluţia exactă tinde către o constantă

$$\hat{x}(t) = x_0 = 1 \tag{8.83}$$

şi $z_0 = 1$. Soluţia numerică se obţine aplicând succesiv relaţia generală (8.54), care se presupune consistentă. În cazul ecuaţiei test (8.80) relaţia generală (8.54) capătă forma:

$$\sum_{i=0}^{p}(a_i + \lambda h b_i)x_{k-i} = 0, \tag{8.84}$$

care este o combinaţie liniară a ultimelor $(p+1)$ valori ale soluţiei numerice. Datorită caracterului liniar al relaţiei (8.84), rezultă că şi diferenţa dintre o soluţie numerică x_k şi perturbata ei z_k, respectiv $e_k = z_k - x_k$ va satisface o ecuaţie asemănătoare:

$$\sum_{i=0}^{p}(a_i + \lambda h b_i)e_{k-i} = 0 \tag{8.85}$$

Presupunem că o perturbaţie iniţială e_0 se propagă în procesul de calcul după o progresie geometrică respectiv

$$e_j = z e_{j-1} \tag{8.86}$$

în care z este coeficientul cu care se amplifică eroarea la fiecare pas. Condiţia ca o metodă să fie absolut stabilă constă în faptul că norma perturbaţiei să nu se mărească de la un pas la altul, deci:

$$|z| \leq 1. \tag{8.87}$$

Substituind relaţia (8.86) în (8.85) rezultă:

$$P(z) = \sum_{i=0}^{p}(a_i + \lambda h b_i)z^{p-i} = 0. \tag{8.88}$$

Polinomul $P(z)$ definit de relaţia (8.88) este numit polinomul caracteristic al metodei iar studiul lui este de mare importanţă în caracterizarea stabilităţii numerice a metodei de integrare. Polinomul caracteristic are gradul p şi poate fi descompus în:

$$P(z) = A(z) + \lambda h B(z), \quad \text{cu} \quad A(z) = \sum_{i=0}^{p} a_i z^{p-i} \quad \text{şi} \quad B(z) = \sum_{i=0}^{p} b_i z^{p-i} \tag{8.89}$$

Pentru început se studiază stabilitatea numerică în condiţiile în care pasul h este suficient de mic. Pentru $h \to 0$, polinomul $P(z)$ se reduce la $A(z)$, numit polinom de stabilitate. Din punctul de vedere al ecuaţiei (8.80), condiţia $\lambda h \to 0$ este echivalentă cu degenerarea $\lambda = 0$ căreia îi corespunde soluţia constantă (8.83). Pentru ca ecuaţia (8.84) să admită în acest caz soluţia constantă este necesar ca:

$$\sum_{i=1}^{p} a_i = 0,$$

ceea ce este adevărat, deoarece reprezintă prima condiţie de consistenţă (8.78). Această condiţie impune ca polinomul de stabilitate $A(z)$ să aibă rădăcina $z_1 = 1$. Această rădăcină a polinomului caracteristic, pentru $\lambda h = 0$ se numeşte **rădăcina principală** şi ea asigură corectitudinea soluţiei. Celelalte rădăcini ale polinomului caracteristic se numesc **rădăcini parazite** şi ele caracterizează propagarea erorilor de calcul. Condiţia necesară pentru ca o metodă multipas să fie numeric stabilă la un pas h suficient de mic este ca toate rădăcinile parazite $z_2, z_3, \ldots, z_{p-1}$ ale polinomului de stabilitate $A(z)$

să aibă modulul mai mic decât cel al rădăcinii principale $z_1 = 1$, deci să fie cuprinse în cercul unitate:

$$|z_j| < 1 \tag{8.90}$$

Se admite şi cazul $|z_j| = 1$ dar cu condiţia să nu existe rădăcini multiple pe cercul unitate (toate rădăcinile de modul unitar trebuie să fie simple). Această condiţie este justificată de inegalitatea (8.87). Pentru un studiu mai atent se va considera metoda consistentă cu doi paşi:

$$(a_0 + \lambda h b_0)x_k + (a_1 + \lambda h b_1)x_{k-1} + (a_2 + \lambda h b_2)x_{k-2} = 0 \tag{8.91}$$

la care polinomul caracteristic este de gradul doi:

$$P(z) = A(z) + \lambda h B(z); \quad A(z) = a_0 z^2 + a_1 z + a_2; \quad B(z) = b_0 z^2 + b_1 z + b_2. \tag{8.92}$$

Polinomul de stabilitate $A(z)$ are rădăcina $z_1 = 1$ şi o alta, z_2, pe care o presupunem diferită de z_1. Dacă se integrează ecuaţia (8.80) cu $\lambda = 0$ soluţia exactă este constantă (8.83). Ecuaţia cu diferenţe finite (8.91) admite o soluţie generală, care este de forma unei combinaţii liniare de serii geometrice, cu raţii date de soluţiile polinomului caracteristic:

$$x_k = c_1 z_1^k + c_2 z_2^k. \tag{8.93}$$

Condiţia iniţială exactă $x_0 = 1, x_1 = 1$ impune valorile:

$$c_1 = 1, \quad c_2 = 0, \tag{8.94}$$

ceea ce ar corespunde unei soluţii numerice egale cu cea exactă. Datorită erorii de rotunjire, este posibil să se obţină:

$$c_1 = 1 \text{ şi } c_2 = \varepsilon \neq 0, \tag{8.95}$$

caz în care soluţia numerică conţine o componentă parazită a cărei valoare iniţială este ε. Dacă modulul rădăcinii parazite $|z_2|$ este subunitar, această componentă nu depăşeşte perturbaţia iniţială ε şi se anulează în timp. Dacă în schimb $|z_2| > 1$, atunci componenta parazită creşte exponenţial în timp şi chiar dacă a plecat de la o valoare iniţială neglijabilă, poate ajunge să depăşească componenta exactă a soluţiei. Dacă rădăcina principală este dublă $z_1 = z_2$, atunci în locul relaţiei (8.93) trebuie folosită relaţia:

$$x_k = c_1 z_1^k + c_2 k z_1^k, \tag{8.96}$$

care pentru $z_1 = z_2 = 1$ devine: $x_k = c_1 + kc_2$ ceea ce evidenţiază o creştere liniară a erorii. Pentru o metodă consistentă condiţia necesară şi suficientă ca rădăcinile unitare să fie simple este ca polinomul $B(z)$ să nu admită rădăcina unitate, respectiv ca [4]:

$$\sum_{i=0}^{p} b_i \neq 0.$$

Dacă se analizează polinomul de stabilitate al metodelor de tip Adams:

$$A(z) = a_0 z^p + a_1 z^{p-1},$$

cu $a_0 = -1$ şi $a_1 = 1$, rezultă că $z_1 = 1$ şi $z_2 = z_3 = \ldots = z_{p-1} = 0$. Deoarece toate rădăcinile parazite ale polinomului de stabilitate sunt nule, rezultă că metodele de tip Adams sunt stabile pentru h suficient de mic. Se poate afirma că din punctul de vedere al stabilităţii numerice la paşi mici, metodele de tip Adams sunt optime deoarece au toate rădăcinile perturbante nule. Polinomul de stabilitate al metodei Milne:

$$A(z) = -(z^2 - 1),$$

are în afara rădăcinii principale $z_1 = 1$ şi rădăcina parazită $z_2 = -1$, deci este stabil pentru $\lambda h \to 0$. Deoarece modulul rădăcinii parazite este unitar, rezultă că erorile de rotunjire la un pas nu sunt atenuate la paşii următori ci se propagă, fără a fi însă amplificate. Metoda Milne prezintă o stabilitate numerică la limită. În schimb, relaţia (8.76) are polinomul de stabilitate:

$$A(z) = -z^2 - 4z + 5 = -(z - 1)(z + 5),$$

la care rădăcina parazită $z_2 = -5$ are modulul supraunitar, ceea ce explică instabilitatea numerică a metodei pentru un pas h oricât de mic. Studiul polinomului $A(z)$ permite caracterizarea comportării metodelor multipas în situaţia limită $\lambda h \to 0$. În cazurile reale, de interes practic, atât λ cât şi h au valori nenule. Metodele multipas care nu amplifică la paşii următori eroarea de rotunjire apărută la pasul curent, în cazul rezolvării ecuaţiei (8.80) cu h impus se numesc *numeric absolut stabile* pentru acea valoare a pasului. Pentru studiul stabilităţii numerice absolute se consideră rădăcinile polinomului caracteristic $P(z)$. Deoarece rădăcinile unui polinom depind continuu de coeficienţii acestuia, una din rădăcini, z_1, este rădăcina principală şi se obţine prin continuitate pornind de la $z_1 = 1$, valoare corespunzătoare parametrului $\lambda h = 0$. În acest caz, rădăcina principală nu mai are valoare unitară ci aproximează valoarea (8.82). Pentru metodele consistente de ordin n, eroarea de aproximare este de tipul $O(\lambda h)^{n+1}$ [9]:

$$z_1 = 1 + \lambda h + \frac{(\lambda h)^2}{2!} + \ldots + \frac{(\lambda h)^n}{n!} + O(\lambda h)^{n+1} \tag{8.97}$$

Dar şi rădăcinile parazite îşi modifică poziţia, fapt ce influenţează în mod diferit propagarea erorii de rotunjire, în funcţie de valoarea parametrului λh. Şi în acest caz soluţia ecuaţiei liniare şi omogene cu diferenţe finite (8.91) este de forma (8.93). Soluţia numerică cea mai apropiată de cea exactă (8.81) se obţine dacă sunt realizate condiţiile (8.94). Prezenţa erorilor la iniţializare sau a erorilor de rotunjire pe parcursul calculului, face ca soluţia numerică să corespundă unor condiţii de tipul (8.95). Componenta perturbatoare $c_2 z_2^k$ trebuie să aibă o valoare care să nu depăşească în timp componenta "exactă" $c_1 z_1^k$, ceea ce impune inegalitatea:

$$|z_j| < |z_1| \tag{8.98}$$

pentru orice rădăcină parazită $z_j, j = 1, 2, \ldots, p-1$. Mulţimea punctelor din planul complex λh, pentru care inegalitatea (8.98) este asigurată poartă numele de *regiune de stabilitate numerică relativă*. În acest domeniu, eroarea chiar dacă se amplifică în timp nu este mare, relativ la soluţia "exactă". În acord cu condiţia (8.87) regiunea de stabilitate numerică absolută corespunde condiţiei $|z_j| \leq 1$ pentru toate rădăcinile polinomului caracteristic.

Considerând spre exemplu, metoda Adams-Bashforth de ordinul doi, la care polinomul caracteristic:

$$P(z) = -z^2 + z + \frac{\lambda h}{2}(3z - 1)$$

are rădăcinile:

$$z_1 = \frac{1}{2}\left[1 + \frac{3}{2}\lambda h + \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h}\right]$$
$$z_2 = \frac{1}{2}\left[1 + \frac{3}{2}\lambda h - \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h}\right]$$

se constată că modulele acestor rădăcini devin egale pentru $\lambda h = -2/3$. Pentru valori reale:

$$\lambda h \geq -2/3, |z_1| \geq |z_2|,$$

metoda este numeric relativ stabilă, iar pentru valori reale:

$$\lambda h < -2/3, |z_1| < |z_2|,$$

metoda este numeric instabilă. Pentru analiza sistemelor instabile ($\lambda > 0$), metoda Adams-Bashforth de ordinul doi poate fi aplicată cu orice pas şi îşi menţine stabilitatea numerică relativă, dar pentru sisteme stabile ($\lambda = -1/\tau < 0$), metoda Adams-Bashforth de ordinul doi este stabilă doar dacă pasul de integrare este suficient de mic

$h < 2\tau/3$. Valorile reale limită ale domeniilor de stabilitate numerică relativă pentru metodele Adams-Bashforth sunt:

$$\begin{aligned} \lambda h &> -0.667, \quad \text{pentru ordinul 2} \\ \lambda h &> -0.358, \quad \text{pentru ordinul 3} \\ \lambda h &> -0.214, \quad \text{pentru ordinul 4.} \end{aligned}$$

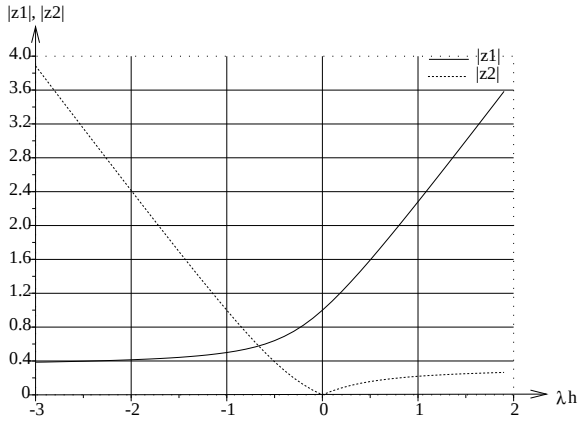


Figura 8.19: Dependenţa de λh a rădăcinilor polinomului caracteristic pentru metoda Adams – Bashforth de ordinul doi

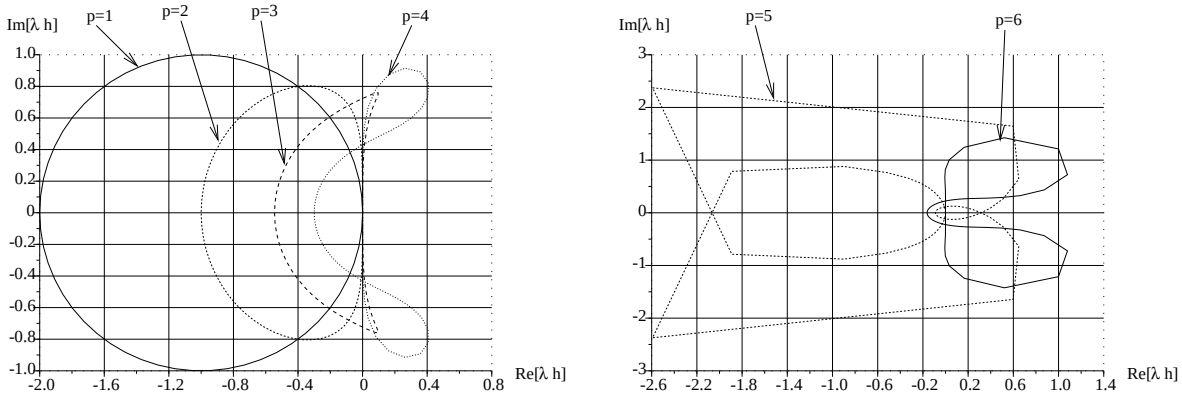


Figura 8.20: Domenii de stabilitate numerică pentru metodele Adams – Bashforth de diferite ordine p

Stabilitatea absolută impune ca atât $|z_1|$ cât şi $|z_2|$ să nu depăşească unitatea, ceea ce corespunde condiţiei $-1 < \lambda h < 0$ (figura 8.19). Domeniile de stabilitate numerică absolută pentru metodele de tip Adams-Bashforth de diferite ordine sunt reprezentate în figura 8.20. Spre deosebire de cazul metodelor de tip Runge-Kutta se constată că pe măsură ce ordinul metodei creşte, atât domeniul de stabilitate relativă cât şi cel de stabilitate absolută se restrâng. Începând de la ordinul trei în sus, domeniul de stabilitate absolută conţine un domeniu din axa imaginară, ceea ce le face potrivite şi

pentru studiul sistemelor autonome cu soluţie oscilantă neamortizată. Metoda Adams-Moulton de ordinul zero (Euler implicită) are polinomul caracteristic:

$$P(z) = -z + 1 + \lambda h z,$$

cu rădăcină unică:

$$z_1 = \frac{1}{1 - \lambda h},$$

iar metoda Adams-Moulton de ordinul unu (metoda trapezelor) are polinomul caracteristic:

$$P(z) = -z + 1 + \frac{\lambda h}{2}(z + 1),$$

cu rădăcina:

$$z_1 = \frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}.$$

Deoarece aceste metode implicite nu au rădăcini parazite, rezultă că sunt numeric relativ stabile pentru orice valoare a pasului de integrare. Stabilitatea numerică absolută este asigurată pentru orice valoare λh reală negativă. În semiplanul drept metoda de ordinul unu este numeric absolut instabilă, pe când metoda de ordinul zero este instabilă doar în interiorul unui cerc cu raza unitate şi centrul în (1, 0). Cu toate că metoda trapezelor oferă o acurateţe mai bună, metoda Euler implicită este folosită des în practică, deoarece are un domeniu de stabilitate mai extins.

Polinomul caracteristic metodei Adams-Moulton de ordinul doi:

$$P(z) = -z^2 + z + \frac{\lambda h}{12}(5z^2 + 8z - 1)$$

are rădăcinile z_1 şi z_2 a căror variaţie faţă de parametrul real λh este reprezentată în figura 8.21.

Rădăcina principală z_1 care aproximează funcţia $e^{\lambda h}$ este depăşită în modul de rădăcina z_2 la valoarea $\lambda h = -1.5$. Modulul rădăcinii parazite $|z_2|$ depăşeşte unitatea la valori $\lambda h < -6$ iar rădăcina principală depăşeşte unitatea pentru $\lambda h > 0$. Rezultă că pe axa reală, regiunea de stabilitate relativă este dată de condiţia $\lambda h \leq -1.5$ iar cea de stabilitate absolută este cuprinsă între limitele $-6 < \lambda h < 0$. Limitele reale ale domeniilor de stabilitate numerică relativă pentru metodele Adams- Moulton sunt:

$$\begin{aligned} &\lambda h > -1.500, \quad \text{pentru } p = 2; \\ &\lambda h > -0.923, \quad \text{pentru } p = 3; \\ &\lambda h > -0.681, \quad \text{pentru } p = 4. \end{aligned}$$

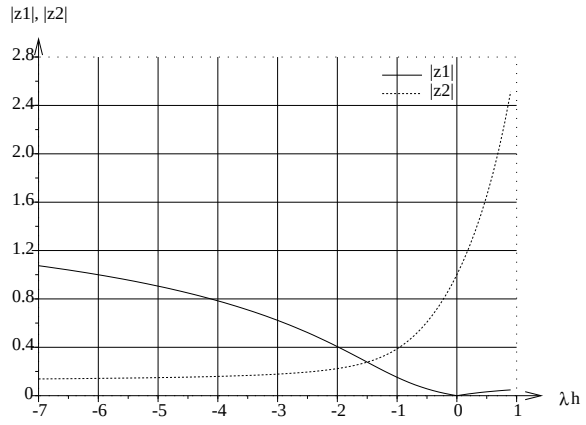


Figura 8.21: Dependenţa de λh a rădăcinilor polinomului caracteristic pentru metoda Adams – Moulton de ordinul doi

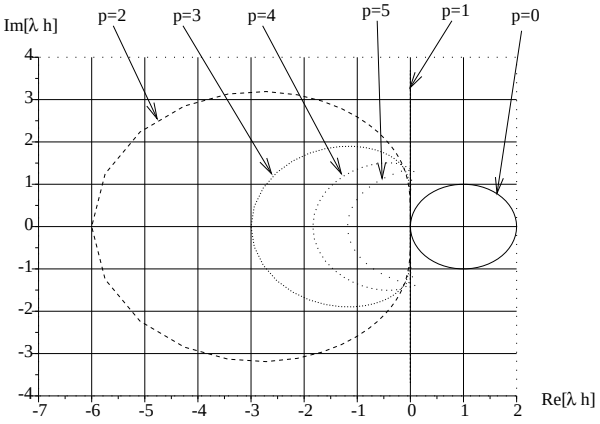


Figura 8.22: Domenii de stabilitate numerică pentru metodele Adams – Moulton de diferite ordine p

Domeniile de stabilitate numerică absolută pentru metodele de tip Adams-Moulton de diferite ordine sunt reprezentate în figura 8.22. Se constată că pe măsură ce ordinul metodei creşte, domeniul de stabilitate se restrânge. Domeniile de stabilitate ale metodelor Adams implicite sunt de câteva ori mai mari decât pentru variantele explicite ale metodelor de acelaşi ordin.

Polinomul caracteristic metodei Milne:

$$P(z) = -z^2 + 1 + \frac{\lambda h}{3}(z^2 + 4z + 1)$$

are două rădăcini, cea principală [26]: $z_1 = e^{\lambda h} + O(h^5)$ şi cea parazită: $z_2 = -e^{-\lambda h/3} + O(h^3)$.

Reprezentând grafic modul în care variază modulul rădăcinilor în funcţie de parametrul λh, presupus real, se evidenţiază că metoda Milne este numeric absolut stabilă doar

în punctul $\lambda h = 0$. Metoda este totuşi numeric relativ stabilă pentru $\lambda h > 0$ ceea ce corespunde sistemelor instabile ($\lambda > 0$).

O teoremă fundamentală privind legătura strânsă între consistenţă, stabilitate numerică (în sensul relaţiei (8.90), privind rădăcinile polinomului de stabilitate) şi convergenţă este demonstrată în [9]. Dacă o metodă multipas este consistentă şi numeric stabilă, atunci ea este convergentă şi reciproc dacă o metodă multipas este convergentă ea este numeric stabilă. Pe baza acestei teoreme metodele de tip Adams şi metoda Milne sunt stabile. Metoda (8.80) însă, este consistentă dar nu este convergentă deoarece nu este stabilă.

8.5 Algoritmul predictor-corector

Algoritmul predictor-corector foloseşte în mod combinat o metodă explicită pentru predicţie:

$$P : x_k^{(0)} = \sum_{i=1}^{p} (a_i' x_{k-i} + h b_i x'_{k-i}) \tag{8.99}$$

pe baza căreia pentru $j = 0$ se evaluează funcţia:

$$E : x_k'^{(j)} = f(x_k^{(j)}, t_k) \tag{8.100}$$

şi apoi în mod iterativ o metodă implicită pentru corecţie:

$$C : x_k^{(j)} = h x_k'^{(j-1)} = \sum_{i=1}^{p} (a_i x_{k-i} + h b_i' x'_{k-i}), \tag{8.101}$$

cu $j = 1, 2, ..., M$.

La fiecare pas de timp t_k sunt parcurse etapele $PECEC.....EC$ sau folosind o notaţie compactă $P(EC)^M$. O variantă îmbunatăţită a algoritmului se încheie cu o etapă de evaluare $P(EC)^M E$. Procesul iterativ continuă până când abaterea între două iteraţii succesive:

$$\varepsilon_j = ||x_k^{(j)} - x_k^{(j-1)}|| \tag{8.102}$$

scade sub o toleranţă impusă erorii. Condiţia (8.102) determină numărul de iteraţii M, efectuate.

Prima problemă care se formulează în legatură cu algoritmul predictor-corector constă în determinarea condiţiilor în care algoritmul iterativ de corecţie este convergent.

Etapele de evaluare (8.100) şi corecţie (8.101) pot fi prezentate compact sub forma:

$$x_k^{(j)} = F(x_k^{(j-1)}), \tag{8.103}$$

în care aplicaţia F este definită de:

$$F(x) = h b_0 f(x, t_k) + \sum_{i-1}^{p} (a_i x_{k-i} + h b_i x'_{k-i}). \tag{8.104}$$

Se constată că limita căutată prin procedeul iterativ de corectare este chiar punctul fix al aplicaţiei $F(x)$.

Pe baza principiului contracţiei, condiţia suficientă ca şirul:

$$x_k^{(0)}, x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(j)}, \ldots$$

definit de relaţia recursivă (8.103) să fie convergent, iar limita sa:

$$x_k = \lim_{j \to \infty} x_k^{(j)}$$

să fie punctul fix al aplicaţiei (8.104) este ca $F(x)$ să fie o contracţie. Cu alte cuvinte, $F(x)$ trebuie să aibă proprietatea Lipschitz cu o constantă subunitară, respectiv să existe o constantă $L < 1$, astfel încât:

$$||F(x') - F(x'')|| \leq L \cdot ||x' - x''||, \tag{8.105}$$

pentru orice pereche x', x'' dintr-o vecinătate a punctului fix x_k.

Presupunând că problema cu condiţii iniţiale este bine formulată, rezultă că funcţia $f(x,t)$ specifică ecuaţiei de rezolvat are proprietatea Lipschitz, cu o constantă care pentru momentul t_k va fi notată L_k, deci:

$$||F(x_k') - F(x_k'')|| = h|b_0| \cdot ||f(x_k', t_k) - f(x_k'', t_k)|| \leq h|b_0| \cdot L_k \cdot ||x_k' - x_k''||. \tag{8.106}$$

Pe baza relaţiei (8.105) se deduce condiţia ca procedeul iterativ să fie convergent:

$$h < \frac{1}{|b_0| \cdot L_k}. \tag{8.107}$$

Condiţia de convergenţă nu impune derivabilitatea funcţiei $f(x,t)$. Dacă totuşi aceasta este derivabilă, norma derivatei, egală cu constanta Lipschitz, trebuie să satisfacă inegalitatea:

$$L_k = ||\frac{\partial f}{\partial x}|| < \frac{1}{h \cdot |b_0|} \tag{8.108}$$

pentru $t = t_k$ şi orice $x_k^{(j)}$ cu $j = 0, 1, 2, \ldots$.

Cu cât constanta de contracţie este mai mică, cu atât algoritmul este mai rapid convergent:

$$\varepsilon_{j+1} = ||x_k^{(j+1)} - x_k^{(j)}|| = ||F(x_k^{(j)}) - F(x_k^{(j-1)})|| \leq L \cdot ||x_k^{(j)} - x_k^{(j-1)}|| \leq L^j \cdot ||x^1 - x^0|| = L^j \cdot \varepsilon_1 \tag{8.109}$$

Dacă inegalitatea (8.107) nu este îndeplinită, atunci procesul iterativ poate fi divergent şi norma abaterii (8.102) creşte de la un pas la altul.

Analizând condiţiile (8.107) sau (8.108) rezultă că pentru a obţine o convergenţă rapidă pasul de timp trebuie să fie suficient de mic, iar dacă pasul creşte, rata de convergenţă

scade. Pentru a optimiza efortul global de calcul trebuie făcut un compromis între valoarea pasului şi rata de convergenţă. Metoda iterativă prezentată are avantajul că abaterea (8.102) constituie un criteriu eficient al convergenţei. Se recomandă ca pasul să fie astfel ales încât să fie necesare cel mult două sau maxim trei iteraţii pentru a obţine toleranţa impusă erorii.

Comparând algoritmul predictor – corector cu algoritmul Runge - Kutta de ordinul patru se constată că primul este de cel puţin două ori mai eficient din punctul de vedere al efortului de calcul, la acelaşi ordin de mărime al erorii soluţiei.

În forma sa clasică, metoda predictor - corector folosea relaţii de tip Milne. Pentru a micşora erorile de trunchiere şi pentru a elimina instabilităţile de calcul, Hamming [7], [6] propune introducerea înainte de etapa de evaluare a unei etape suplimentare numită *de modificare.* În versiunea Hamming este esenţial ca predictorul şi corectorul să ofere acelaşi ordin al erorii de trunchiere, chiar dacă numărul de paşi este diferit în cele două relaţii. Modificarea operează asupra corectorului, folosind diferenţa dintre predictor şi corector de la pasul anterior şi urmăreşte anularea erorii de trunchiere. Pentru relaţiile de ordinul patru se propune:

$$\begin{aligned}
&P : p_k = x_{k-4} + \tfrac{4h}{3}(2x_{k-1} - x_{k-2} + 2x_{k-3});\\
&M : m_k = p_k - \tfrac{112}{121}(p_{k-1} - c_{k-1});\\
&E : m_k = f(m_k, t_k);\\
&C : c_k = \tfrac{1}{8}[9x_{k-1} - x_{k-2} + 3h(m_k + 2x_{k-1} - x_{k-2})].
\end{aligned}$$

În final, pentru soluţia de la momentul t_k se adoptă valoarea:

$$x_k = c_k + \frac{9}{121}(p_k - c_k).$$

În versiunea acestui algoritm implementată în [6] nu se efectuează iteraţii asupra valorii prezise, în schimb se adoptă o strategie a modificării pasului de timp, bazată pe diferenţa dintre predictor şi corector la momentul de timp k:

$$\varepsilon_k = \sum_{i=1}^{n} g_i \cdot |p_k^{(i)} - c_k^{(i)}|.$$

în care indicele i este asociat variabilei de stare curente, iar g_i este ponderea dată acestuia în calculul erorii.

Se deosebesc patru cazuri:

- dacă $\varepsilon_k < \varepsilon_m$, atunci pasul de timp se dublează;
- dacă $\varepsilon_m < \varepsilon_k \leq \varepsilon_M$, atunci valoarea pasului de timp se menţine constantă;
- dacă $\varepsilon_k > \varepsilon_M$, atunci pasul de timp se înjumătăţeşte;

- dacă $\varepsilon_k > \varepsilon_{\max} \geq \varepsilon_M$, atunci soluţia se recalculează fără avansarea pasului.

Pentru a permite modificarea pasului, soluţia este memorată pe 5 paşi. La înjumătăţire sunt folosite relaţii de interpolare care asigură erori de cel puţin acelaşi ordin ca relaţiile de integrare.

8.6 Reprezentarea canonică a metodelor multipas

Prezentarea algoritmului predictior - corector într-o formă matriceală, numită *reprezentare canonică* urmăreşte mai multe aspecte:

- unificarea teoriei generale a metodelor cu mai mulţi paşi;
- programarea mai eficientă a algoritmului;
- facilitarea modificării ordinului şi a pasului.

Pentru a simplifica notaţiile se consideră o singură ecuaţie diferenţială scalară:

$$\frac{dx}{dt} = f(x,t) \tag{8.110}$$

cu condiţia iniţială $x(t_0) = x_0 \in \mathbf{R}$. Valorile soluţiei numerice şi ale pantei acesteia în ultimele p noduri ale unei reţele uniforme de discretizare $t_{k-p}, t_{k-p+1}, \ldots, t_{k-2}, t_{k-1}$ pot fi considerate componentele unui "vector soluţie numerică":

$$y_{k-1} = [x_{k-1}, x_{k-2}, \ldots, x_{k-p}, hx'_{k-1}, hx'_{k-2}, \ldots, hx'_{k-p}]^T. \tag{8.111}$$

Scopul unei metode multipas constă în determinarea vectorului y_k pornind de la y_{k-1} şi avansând cu un pas de timp, în vederea calculului soluţiei numerice a ecuaţiei (8.110). Acest proces poate fi aplicat succesiv pentru a calcula $y_1, y_2, \ldots, y_m$, presupunând pe y_0 cunoscut. Problema calculului iniţializării y_0, numită şi problema "startului", face ca metodele multipas să fie evitate. Totuşi, pentru probleme mari, care solicită acuratete a soluţiei numerice, eficienţa oferită de metodele multipas este net superioară metodelor cu un pas. O metodă uzuală în rezolvarea problemei startului constă în utilizarea metodelor cu un pas, ca de exemplu metoda Runge - Kutta, pentru a parcurge primii $(p-1)$ paşi şi pentru a calcula y_0. La start trebuie acordată o atenţie deosebită controlului erorii, deorece o iniţializare inexactă, afectată de erori grosolane, poate influenţa negativ exactitatea soluţiei numerice de la paşii următori. În prima etapă a calculului, la un nou pas de timp se acceptă procedeul de predicţie, bazat pe relaţia (8.99), prin care valoarea iniţială $y_k^{(0)}$ a vectorului y_k se obţine printr- o transformare liniară pornind de la vectorul y_{k-1}. Prima componentă $x_k^{(0)}$ se calculează cu relaţia (8.99) iar următoarele $(p-1)$ se obţin printr-o translaţie. Pentru calculul pantei $hx_k'^{(0)}$ se utilizează relaţia (8.101), în care $x_k^{(1)}$ se consideră dat de (8.99), deci:

$$hx'_k = \sum_{k=1}^{p} [c_i x_{k-i} + h d_i x'_{k-i}], \tag{8.112}$$

cu $c_i = (a'_i - a_i)b_0$, $d_i = (b'_i - b_i)b_0$, iar următoarele componente se obţin tot prin translaţie. Acest mod de scriere pentru $x_k^{\prime(0)}$ prezintă avantaje ulterioare în calculul corectorului. În consecinţă, etapa poate fi reprezentată matriceal sub forma:

$$y_k^{(0)} = Y \cdot y_{k-1}, \tag{8.113}$$

sau dezvoltat:

$$\underbrace{\begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \\ \dots \\ x_{k-p+1} \\ hx'_{k-1} \\ hx'_{k-2} \\ \dots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(0)}} = \underbrace{\begin{bmatrix} a'_1 & a'_2 & \dots & a'_{p-1} & a'_p & b'_1 & b'_2 & \dots & b'_{p-1} & b'_p \\ 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ c_1 & c_2 & \dots & c_{p-1} & c_p & d_1 & d_2 & \dots & d_{p-1} & d_p \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_{Y} \cdot \underbrace{\begin{bmatrix} x_{k-1} \\ x_{k-2} \\ x_{k-3} \\ \dots \\ x_{k-p} \\ hx'_{k-1} \\ hx_{k-2} \\ hx_{k-3} \\ \dots \\ hx_{k-p} \end{bmatrix}}_{y_{k-1}}$$

Procedeul de corecţie se bazează pe relaţiile (8.100), (8.101). Analizând relaţia (8.101), rezultă că termenul care conţine simbolul de sumă rămâne neschimbat în timpul iteraţiei, deci prin scăderea relaţiilor pentru două iteraţii succesive, se obţine:

$$x_k^{(j+1)} = x_k^{(j)} + hb_0[x_k^{\prime(j)} - x_k^{\prime(j-1)}]. \tag{8.114}$$

Această relaţie se poate aplica şi pentru cazul $j = 0$ dacă se adoptă formal pentru $x_k^{\prime(-1)}$ valoarea $x_k^{\prime(0)}$ dată de relaţia (8.112).

Sub formă matriceală, relaţia (8.114) are forma:

$$y_k^{(j+1)} = y_k^{(j)} + F(y_k^{(j)}) \cdot c_y \tag{8.115}$$

cu $j = 0, 1, 2, \dots, (M-1)$, sau dezvoltat:

$$\underbrace{\begin{bmatrix} x_k^{j+1} \\ x_{k-1} \\ \vdots \\ x_{k-p+1} \\ hx_k^{\prime(j+1)} \\ hx'_{k-1} \\ \vdots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(j+1)}} = \underbrace{\begin{bmatrix} x_k^{(j)} \\ x_{k-1} \\ \vdots \\ x_{k-p+1} \\ hx_k^{\prime(j)} \\ hx'_{k-1} \\ \vdots \\ hx'_{k-p+1} \end{bmatrix}}_{y_k^{(j)}} + \underbrace{h\left[f(x_k^{(j)}, t_k) - x_k^{\prime(j)}\right]}_{F(y_k^{(j)})} \cdot \underbrace{\begin{bmatrix} b_0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{c_y}$$

În concluzie, metoda predictor - corector sub formă matriceală:

$$\begin{aligned} y_k^{(0)} &= Y \cdot y_{k-1} \\ y_k^{(j+1)} &= y_k^{(j)} + F(y_k^{(j)}) \cdot c_y, j = 0, 1, \ldots, (M-1) \end{aligned} \tag{8.116}$$

este caracterizată de o matrice de predicţie Y, de dimensiune $2p \times 2p$ care conţine toate informaţiile privind atât coeficienţii predictorului, cât şi ai corectorului (cu excepţia lui b_0). Această matrice înmulţeşte o singură dată vectorul y_{k-1} şi nu depinde de funcţia $f(x,t)$, predicţia fiind invariantă la ecuaţia de rezolvat.

Corecţia reprezintă un procedeu iterativ iar la fiecare iteraţie se modifică doar două componente din vectorul soluţie $y_k^{(j)}$ şi anume prima şi componenta de rang $(p+1)$, care corespund valorilor $x_k^{(j)}$ şi $x_k'^{(j)}$. Pentru aceasta se evaluează funcţia $f(x,t)$ la momentul curent t_k, pentru ultima valoare a soluţiei numerice $x_k^{(j)}$. Valoarea soluţiei numerice (prima componentă a vectorului $y_{k-1}^{(j)}$) se incrementează cu:

$$b_0 h[f(x_k^{(j)}, t_k) - x_k'^{(j)}], \tag{8.117}$$

iar valoarea pantei soluţiei x_k (componenta de rang $(p+1)$ a vectorului $y_k^{(j)}$) se actualizează la valoarea nou calculată:

$$x_k^{(j+1)} = h f(x_k^{(j)}, t_k). \tag{8.118}$$

La prima iteraţie, pentru $j = 0$, valoarea $x_k'^{(0)}$ necesară relaţiei (8.117) este preluată din vectorul $y_k^{(0)}$, de la componenta $(p+1)$, a cărei valoare este dată de (8.112) . Prin substituţie în (8.117) se obţine:

$$x_k^{(1)} = x_k^{(0)} + b_0 \left[h f(x_k^{(0)}, t_k) - \sum_{i=1}^{p} (u x_{k-i} + h d_i x'_{k-i}) \right],$$

sau ţinând seama de expresiile coeficienţilor c_i, d_i precum şi de expresia predictorului $x_k^{(0)}$, rezultă relaţia (8.101), particularizată pentru $j = 1$, care dă prima iteraţie a corectorului.

Acest mod de scriere matriceală a corectorului evidenţiază că el este determinat în principal de funcţia $f(x,t)$ şi de vectorul c_y, a cărui componentă principală este coeficientul termenului implicit b_0.

În continuare se va exemplifica scrierea matriceală a metodei predictor-corector pentru cazul utilizării relaţiilor de tip *Adams* pe trei paşi (8.68), (8.69). Coeficienţii acestor relaţii au valorile:

$$\begin{array}{lll} a_1' = 1, & a_2' = 0, & a_3' = 0; \\ b_1' = 23/12, & b_2' = -16/12, & b_3' = 5/12; \\ a_1 = 1, & a_2 = 0, & a_3 = 0; \\ b_1 = 19/24, & b_2 = -5/24, & b_3 = 1/24 \end{array}$$

şi $b_0 = 9/24$, care permite calculul coeficienţilor:

$$\begin{array}{lll} c_1 = 0, & c_2 = 0, & c_3 = 0; \\ d_1 = 3, & d_2 = -3, & d_3 = 1, \end{array}$$

ceea ce corespunde matricei de predicţie:

$$Y = \begin{bmatrix} 1 & 0 & 0 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -3 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

şi matricei de corecţie:

$$c_y = \begin{bmatrix} \frac{3}{8} & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T.$$

Deorece din vectorul soluţiilor:

$$y_k = \begin{bmatrix} x_k & x_{k-1} & x_{k-2} & hx'_k & hx'_{k-1} & hx'_{k-2} \end{bmatrix}^T,$$

componentele x_{k-1} şi x_{k-2} nu sunt utilizate nici în procesul de predicţie nici în cel de corecţie, acestea pot fi eliminate şi odată cu ele se elimină şi elementele corespunzătoare din matricele Y (liniile şi coloanele 2, 3) şi c_y (liniile 2, 3):

Aceasta corespunde relaţiei de predicţie:

$$\begin{bmatrix} x_k^{(0)} \\ hx_k'^{(0)} \\ hx_{k-1} \\ hx_{k-2} \end{bmatrix} = \begin{bmatrix} 1 & \frac{23}{12} & -\frac{16}{12} & \frac{5}{12} \\ 0 & 3 & -3 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ hx'_{k-1} \\ hx'_{k-2} \\ hx'_{k-3} \end{bmatrix} \tag{8.119}$$

şi relaţiei de corecţie cu $j = 0, 1, 2, \ldots, (M-1)$:

$$\begin{bmatrix} x_k^{(j+1)} \\ hx_k'^{(j+1)} \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} = \begin{bmatrix} x_k^{(j)} \\ hx_k^{(j)} \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} + F \cdot \begin{bmatrix} \frac{3}{8} \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{8.120}$$

unde:

$$F = h\left[f(x_k^{(j)}, t_k) - x_k'^{(j)}\right], \tag{8.121}$$

relaţii în concordanţă cu cele scalare.

Constatăm că prin reprezentarea matriceală datele au căpătat o structură simplă şi elegantă. Algoritmul predictor-corector este caracterizat prin matricea de predicţie Y, de dimensiune 4×4 şi vectorul de corecţie c_y de dimensiune 4. Soluţia este reprezentată la pasul curent de un vector cu patru elemente, din care primul dă valoarea soluţiei iar al doilea dă panta ei de variaţie în timp, multiplicată cu pasul de integrare h.

După iniţializarea corespunzătoare a vectorului soluţie y_2 la pasul de timp $k = 2$ în vederea calculului soluţiei numerice la pasul $k = 3$, se aplică relaţia de predicţie (8.119) iar asupra valorilor obţinute în vectorul y_k se aplică iterativ procedura de corecţie (8.120), care presupune evaluarea funcţiei $f(x, t)$ la fiecare iteraţie, conform relaţiei

(8.121). Iteraţiile se opresc atunci când norma abaterii între două iteraţii succesive scade sub toleranţa impusă. Valoarea obţinută se consideră soluţia numerică de la pasul k: $y_k = y_k^{(M)}$ şi apoi se incrementează pasul de timp, iar algoritmul predictor-corector se reia până la atingerea timpului final.

Algoritmul prezentat se poate generaliza uşor pentru cazul rezolvării unor sisteme de n ecuaţii diferenţiale. În acest caz elementele vectorului soluţie y_k sunt la rândul lor vectori cu n componente, relaţiile (8.119), (8.120) aplicându-se în mod independent pentru fiecare componentă în parte. Dar în relaţia (8.121), în evaluarea funcţiei $f(x,t)$ pot interveni toate componentele soluţiei la pasul curent şi la iteraţia curentă.

8.7 Metode cu valori multiple

În reprezentarea matriceală canonică a metodei predictor-corector, soluţia la un pas de integrare este caracterizată printr-un vector cu $2p$ componente:

$$y_k = \left[x_k, x_{k-1}, \ldots, x_{k-p}, hx_k, hx_{k-1}, \ldots, hx'_{k-p}\right]^T, \tag{8.122}$$

din care primele p constituie valorile soluţiei numerice pe ultimii paşi de integrare, iar ultimele p componente reprezintă pantele soluţiei la acei paşi, pante multiplicate cu valoarea pasului de timp h.

Această reprezentare a soluţiei nu este cea mai potrivită din punctul de vedere al efortului de calcul sau al erorii de rotunjire. În aceste cazuri se poate dovedi mai potrivită o reprezentare a soluţiei de la un moment de timp prin $2p$ combinaţii liniare ale componentelor canonice ale vectorului 8.122. Se presupune că aceste combinaţii sunt organizate într-un nou vector notat z_k. Trecerea de la vechiul vector y_k la noul vector este asigurată de o transformare liniară:

$$z_k = T \cdot y_k, \tag{8.123}$$

reprezentată de o matrice pătrată de dimensiune $2p \times 2p$. Pentru ca noua reprezentare să fie echivalentă celei vechi este necesar ca transformarea liniară (8.123) să fie inversabilă, deci matricea T să fie nesingulară. În aceste condiţii trecerea inversă este asigurată de relaţia:

$$y_k = T^{-1} z_k. \tag{8.124}$$

Folosind noua reprezentare echivalentă, forma canonică a metodei predictor-corector devine:

$$\begin{aligned} z_k^{(0)} &= T y_k^{(0)} = TY y_{k-1} = TYT^{-1} z_{k-1}; \\ z_k^{(j+1)} &= T y_k^{(j)} + F\left(y_k^{(j)}\right) T c_y = z_k^{(j)} + F\left(T^{-1} z_k^{(j)}\right) c_y, \end{aligned} \tag{8.125}$$

sau folosind relaţiile:

$$Z = TYT^{-1}; \quad G\left(z_k\right) = F\left(T^{-1} z_k\right); \quad c_z = T c_y, \tag{8.126}$$

se obţine în final:

$$z_k^{(0)} = Z \cdot z_{k-1}; \quad z_k^{(j+1)} = z_k^{(j)} + G\left(z_k^{(j)}\right) c_z. \tag{8.127}$$

Relaţiile (8.127) reprezintă forma matriceală canonică pentru vectorul echivalent z_k a metodei predictor-corector. Deoarece componentele vectorului z_k, soluţie de la pasul t_k nu mai reprezintă valorile soluţiei pe ultimii paşi, metoda generată de relaţiile (8.127) nu mai poate fi numită o metodă multipas. Ea se va numi în continuare o *metodă cu valori multiple ale soluţiei.*

Metodele cu valori multiple generalizează metodele cu mai mulţi paşi. Dacă matricea de transformare T este matricea unitate, atunci metoda cu valori multiple devine o metodă multipas.

Trebuie menţionat că o transformare de echivalenţă nu afectează eroarea de trunchiere sau stabilitatea metodei, singurele aspecte care se pot modifica sunt efortul de calcul şi ordinul erorii de rotunjire. Pe un calculator ideal, fară eroare de rotunjire, două metode echivalente trebuie să obţină rezultate identice pentru aceeaşi problemă.

Un exemplu îl constituie utilizarea în locul vectorului canonic $y_k = \left[x_k, hx'_k, hx'_{k-1}, hx'_{k-2}\right]$ a vectorului echivalent $z_k = \left[x_k, hx'_k, x_{k-1}, hx'_{k-1}\right]$. Componentele acestuia reprezintă valoarea soluţiei pe ultimii doi paşi dar deoarece el este echivalent cu o reprezentare pe trei paşi nu se poate spune că metoda generată este cu doi paşi, ci mai corect că este o metodă cu patru valori.

8.7.1 Reprezentarea Nordsieck

În continuare se va prezenta o tehnică de alegere a transformării T, datorată lui Nordsieck, care facilitează modificarea pasului de timp h, pe parcursul integrării.

În reprezentarea Nordsieck, soluţia este caracterizată la pasul curent de vectorul echivalent:

$$z_k = \left[x_k;\ hx'_k;\ \frac{h^2}{2}x''_x;\ \ldots;\ \frac{h^{2p+1}}{(2p+1)!}x_k^{(2p+1)}\right]^T , \tag{8.128}$$

care conţine drept componente primii 2p termeni ai seriei Taylor asociate soluţiei exacte.

În acest caz, valorile multiple constau de fapt în valorile derivatei funcţiei la momentul curent, cu ordine începând de la *zero* până la $(2p+1)$, derivate ponderate corespunzător. Cu toate că o astfel de reprezentare este echivalentă formei canonice multipas, ea nu se mai poate numi aşa, deoarece nu mai conţine în mod explicit informaţii privind comportarea soluţiei pe mai mulţi paşi de timp.

Pentru a obţine forma canonică a metodei predictor-corector în reprezentarea Nordsieck trebuie determinată matricea T a transformării de echivalenţă. În acest scop se consideră un polinom arbitrar de gradul trei:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

ale cărui derivate sunt:

$$\begin{aligned} x'(t) &= a_1 + 2a_2 t + 3a_3 t^2; \\ x''(t) &= 2a_2 + 6a_3 t; \\ x'''(t) &= 6a_3. \end{aligned}$$

Pe o reţea de discretizare uniformă cu $t_k = 0$, $t_{k-1} = -h$, $t_{k-2} = -2h$, componentele vectorului canonic sunt:

$$\begin{aligned} x_k &= a_0; \\ x'_k &= a_1; \\ x'_{k-1} &= a_1 - 2a_2h + 3a_3h^2; \\ x'_{k-2} &= a_1 - 4a_2h + 12a_3h^2. \end{aligned}$$

Pentru acest caz, componentele vectorului Nordsieck sunt:

$$\begin{aligned} x_k &= a_0; \\ x'_k &= a_1; \\ x'_{k-1} &= a_1 - 2a_2h + 3a_3h^2; \\ x'_{k-2} &= a_1 - 4a_2h + 12a_3h^2. \end{aligned}$$

Matricea de transformare are inversa dată de:

$$y_k = \begin{bmatrix} x_k \\ hx'_k \\ hx'_{k-1} \\ hx'_{k-2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -2 & 3 \\ 0 & 1 & -4 & 12 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ ha_1 \\ h_2^a \\ h^3a_3 \end{bmatrix} \tag{8.129}$$

Prin inversarea matricei T^{-1} se obţine:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{3}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{6} & \frac{-1}{3} & \frac{1}{6} \end{bmatrix}. \tag{8.130}$$

Folosind forma canonică a matricelor Y şi c_y dată de (8.119), (8.120) precum şi relaţiile de transformare (8.126), rezultă:

$$Z = TYT^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad c_z = Tc_y = \begin{bmatrix} 3/8 \\ 1 \\ 3/4 \\ 1/6 \end{bmatrix}. \tag{8.131}$$

Deoarece atât vectorul canonic cât şi cel Nordsieck conţin componentele x_k, hx'_k şi aceastea ocupă aceleaşi poziţii (1) şi (2), rezultă că funcţia F nu suferă practic modificări:

$$G(z_k) = F(y_k) = h\left[f\left(x_k^{(j}, t_k\right) - x_k^{\prime(j)}\right]. \tag{8.132}$$

Matricea de predicţie Z are forma triunghiular superioară iar elementele acestei matrice se recunosc a fi cele din triunghiul Pascal:

$$z_{i+1,j+1} = C_i^j = \frac{j!}{(j-i)! \cdot i!}, \quad 0 \le i \le j < 4. \tag{8.133}$$

Acest lucru nu este întâmplător deoarece relaţia de predicţie exprimă de fapt vectorul Nordsieck de la pasul $t = t_{k+1}$:

$$y_{k+1} = \left[x_{k+1};\ hx'_{k+1};\ \ldots;\ \frac{h^p x^{(p)}_{k+1}}{p!} \right]^T$$

şi vectorul Nordsieck de la momentul de timp $t = t_k$:

$$y_k = \left[x_k;\ hx'_k;\ \ldots;\ \frac{h^p x^{(p)}_k}{p!} \right]^T$$

Folosind dezvoltarea în serie Taylor trunchiată a soluţiei în origine se obţine:

$$x(t) = \sum_{j=0}^{p} \frac{x_k^{(j)}}{j!} t^j = \sum_{j=0}^{p} \frac{z_k^{(j)}}{h^j} t^j,$$

în care s-a notat cu $z_k^{(j)}$ componenta j a vectorului Nordsieck de la pasul k. Derivata de ordin i a acestei funcţii este:

$$x^{(i)}(t) = \sum_{j=0}^{p} \frac{j!}{(j-i)!} \frac{z_k^{(j)}}{h^j} t^{j-i}, \quad \text{cu} \quad i \le j,$$

sau particularizând pentru $t = h$, rezultă:

$$x^{(i)}_{k+1} = \sum_{j=0, j\ge i}^{p} \frac{j! h^{-i}}{(j-i)!} z_k^{(j)},$$

ceea ce permite calculul componentei i a vectorului Nordsieck la momentul $k+1$:

$$z^{(i)}_{k+1} = \frac{h^i}{i!} x^{(i)}_{k+1} = \sum_{j=0, j\ge i}^{p} \frac{j!}{(j-i)! \cdot i!} z_k^{(j)}.$$

Coeficienţii relaţiei de legătură formează un triunghi Pascal iar relaţia (8.133) este valabilă în cazul general:

$$z_{i+1,j+1} = C_i^j = \frac{j!}{(j-i)! \cdot i!}, \tag{8.134}$$

cu $0 \le i \le j < p$.

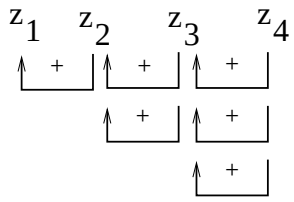
Deoarece matricea Z are mai multe elemente nule decât matricea Y s-ar părea că efortul de calcul necesar predicţiei este mai mare. În realitate, dacă se ia în considerare relaţia:

$$C_i^j + C_{i+1}^j = C_{i+1}^{j+1},$$

rezultă că produsul matricelor $Z \cdot z_{k-1}$ se poate realiza doar prin operaţii de adunare. Produsul dintre triunghiul Pascal şi vectorul Nordsieck:

$$z_{k-1} = [z_1, z_2, \ldots, z_p]^T$$

constă în operaţiile (exemplificate pentru $p = 4$):



$$
\begin{aligned}
z_j &= z_j + z_{j+1}; \quad j = p-1 \to 1 \\
z_j &= z_j + z_{j+1}; \quad j = p-1 \to 2 \\
&\ldots\ldots\ldots\ldots \\
z_{p-1} &= z_{p-1} + z_p
\end{aligned}
$$

deci în total un număr de $p(p-1)/2$ adunări.

Segmentul de cod pseudocod care asigură aceste operaţii pentru cazul a n ecuaţii şi p componente în vectorul Nordsieck de dimensiune $z(p, n)$ este:

```
pentru j = 2, p
        pentru j1 = j, p
                j2 = p - j1 + j -1
                pentru i = 1, n
                        z(j2, i) = z(j2, i) + z(j2 + 1, i)
```

Eliminarea înmulţirilor din etapa de predicţie scurtează sensibil timpul de calcul asociat acestei etape.

Un alt avantaj al reprezentării Nordsieck constă în faptul că utilizatorul are la dispoziţie la fiecare pas de integrare valoarea soluţiei şi a primelor sale derivate. Cel mai evident avantaj al acestei metode constă în uşurinţa cu care se poate modifica pasul de timp pe parcursul integrării.

8.7.2 Controlul automat al ordinului şi mărimii pasului în metodele cu valori multiple

Spre deosebire de cazul metodelor cu un pas, la care creşterea ordinului determină sporirea efortului de calcul prin mărirea numărului de evaluări ale funcţiei $f(x, t)$, în cazul metodelor cu valori multiple modificarea pasului nu influenţează sensibil efortul de calcul.

Faptul că micşorarea mărimii pasului determină scăderea erorii de trunchiere cu atât mai rapid cu cât metoda are ordin mai înalt poate duce la ideea că metodele de ordin ridicat sunt mai bune decât cele de ordin scăzut. În realitate, metodele de ordin ridicat au un domeniu de stabilitate mai restrâns, ceea ce face ca la un pas dat probabilitatea de apariţie a instabilităţilor numerice să fie mai mare la metodele de ordin mai înalt.

În metodele de tip predictor-corector, la care numărul de iteraţii corectoare este suficient de mare pentru ca soluţia la pasul k să satisfacă practic exact ecuaţia corectorului, eroarea de trunchiere este dată de ordinul erorii corectorului. Dacă numărul de iteraţii este mic iar soluţia numerică adoptată este relativ depărtată de punctul fix al relaţiei de corecţie, atunci eroarea de trunchiere are ordinul erorii predictorului. Aceleaşi afirmaţii sunt adevărate şi cu privire la stabilitatea numerică a metodelor predictor-corector.

După cum s-a arătat, metodele Adams-Bashforth de ordinul p au o eroare locală de trunchiere de tipul $O(h^{p+1})$, care poate fi exprimată sub forma:

$$\varepsilon_k = C_k \cdot h^{p+1}.$$

Eroarea globală la momentul de timp t_m nu trebuie să depăşească toleranţa impusă:

$$\varepsilon_k \frac{t_m}{h} = C_k \cdot h^p \cdot t_m \leq \varepsilon_{\max}.$$

Dacă se reprezintă modul de variaţie a erorii globale medii pe pas $\epsilon_{\max} = \epsilon_k/h$ în funcţie de mărimea pasului pentru diferite ordine ale metodei, se obţin grafice de tipul celor din figura 14.

Figura 14 Eroarea globală medie

Din analiza acestei figuri se constată că pentru o eroare impusă, la fiecare ordin pasul are o valoare maximă. Ordinul care permite pasul maxim depinde de mărimea erorii impuse. La erori mici, cel mai mare pas maxim îl au metodele de ordin ridicat, în schimb la erori mari sunt preferate metodele de ordin mic. În acest fel axa erorilor se divide în subdomenii, cu proprietatea că pe fiecare subdomeniu este recomandat un anumit ordin optim. Modul în care variază eroarea de trunchiere în funcţie de mărimea pasului, cu condiţia ca metoda să aibă ordinul optim, este dat de curba anvelopă la dreapta pentru familia de grafice din figura 14.

Din punctul de vedere al programării, micşorarea ordinului cu o unitate constă în principiu în anularea ultimelor linii din matricele Y, Z, c_y sau c_z şi a ultimei coloane din matricele Y şi Z. Creşterea ordinului presupune adăugarea la loc a elementelor eliminate anterior.

Cu toate că ultimul element din vectorul soluţie y_k nu mai este necesar după micşorarea ordinului, el este memorat în continuare pentru cazul în care ordinul va fi din nou majorat. Acest lucru presupune ca la start iniţializarea vectorului y_0 să fie făcută pentru cazul corespunzător ordinului maxim. În implementările practice, cu control automat al ordinului, acesta variază între 1 şi maxim 6.

Pentru a asigura salvarea ultimelor elemente din vectorul y_k, la trecerea la pasul următor, în zona anulată a matricelor Y sau Z este necesar ca elementele subdiagonale să fie unitare. De exemplu pentru matricea (8.131):

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

În cazul metodelor multipas, modificarea mărimii pasului de integrare întâmpină dificultăţi mult mai mari decât în cazul metodelor cu un pas. Dacă pentru pasul de timp h, la momentul t_k, vectorul soluţie este:

$$y_k = \left[x_k,\ x_{k-1}, \ldots, x_{k-p},\ h'x'_k,\ h'x'_{k-1}, \ldots,\ h'x'_{k-p}\right] \tag{8.135}$$

şi se doreşte modificarea pasului la valoarea $h' = \alpha \cdot h$, pentru calculele următoare este necesar calculul vectorului:

$$\hat{y}_k = \left[\hat{x}_k,\ \hat{x}_{k-1}, \ldots, \hat{x}_{k-p},\ h'\hat{x}'_k,\ h'\hat{x}'_{k-1}, \ldots,\ h'\hat{x}'_{k-p}\right], \tag{8.136}$$

în care doar două componente se obţin direct din (8.135: $\hat{x}_k = x_k$ şi $h'\hat{x}_k = \alpha \cdot h \cdot x'_k$. Celelalte componente reprezintă valori ale soluţiei numerice şi ale pantei acesteia, valori ale variabilei independente t, care nu sunt noduri ale reţelei iniţiale de discretizare. Dacă se doreşte micşorarea pasului de discretizare ($\alpha < 1$), atunci aceste valori se pot calcula prin interpolare. Din punct de vedere matriceal, realizarea interpolării este echivalentă cu transformarea liniară:

$$\hat{y}_k = C\left(\alpha\right) \cdot y_k \tag{8.137}$$

în care $C(\alpha)$ este o matrice pătrată ale cărei elemente depind de α. Dacă se doreşte mărirea pasului de discretizare ($\alpha > 1$), atunci valorile calculate prin extrapolare sunt susceptibile de erori de aproximare mari. Pentru a evita erorile de extrapolare este preferabilă extinderea vectorului soluţie pentru memorarea rezultatelor pe un număr de paşi mai mare decât cel necesar în procesul de predicţie. Dificultăţile întâlnite fac ca utilizatorii metodelor multipas să fie descurajaţi la modificarea mărimii pasului.

Nu acelaşi lucru se poate afirma despre metodele cu valori multiple. Metoda Nordsieck a fost special concepută pentru a facilita modificarea pasului. Deoarece derivatele soluţiei la momentul curent nu se modifică prin modificarea pasului rezultă că trecerea de la vectorul Nordsieck pentru momentul t_k şi pasul h:

$$z_k = \left[x_k,\ hx'_k,\ \frac{h^2}{2}x''_k, \ldots, \frac{h^p}{p!}x_k^{(p)}\right]^T \tag{8.138}$$

la vectorul corespunzător aceluiaşi moment, dar cu pasul $h' = \alpha h$:

$$\hat{z}_k = \left[x_k,\ h'x'_k,\ \frac{h'^2}{2}x''_k, \ldots, \frac{h'^p}{p!}x_k^{(p)}\right]^T \tag{8.139}$$

se realizează prin multiplicarea componentelor cu $1, \alpha, \alpha^2, \ldots$ respectiv prin aplicarea transformării (8.137), în care matricea $C(\alpha)$ este diagonală:

$$C(\alpha) = \begin{bmatrix} 1 & & & & 0 \\ & \alpha & & & \\ & & \alpha^2 & & \\ & & & \ddots & \\ 0 & & & & \alpha^p \end{bmatrix}. \tag{8.140}$$

Usurinţa cu care se poate modifica pasul în metoda Nordsieck permite adaptarea în permanenţă a mărimii acestuia la valoarea optimă, corespunzătoare erorii impuse. Modificarea pasului de integrare poate fi făcută urmărind abaterea dintre predictor şi corector, abaterea dintre două iteraţii succesive la corector sau reducerea la doi a numărului de iteraţii corectoare. Toate aceste criterii nu urmăresc direct eroarea de trunchiere a metodei şi pot da indicaţii false pentru modificarea pasului. Gear propune un algoritm de control al ordinului şi mărimii pasului, algoritm bazat pe controlul erorii de trunchiere. Eroarea locală de trunchiere cu metoda Adams-Moulton de ordin p poate fi exprimată sub forma:

$$\varepsilon_t = \left[C_p x^{(p+2)}(\tau)\right] h^{p+2} \tag{8.141}$$

dependenţa valorii derivate de ordin $(p+2)$ a soluţiei într-un punct cuprins în intervalul (t_k, t_{k+1}), cu constantele C_p dependente de ordin:

$$\begin{aligned} C_0 &= -\frac{1}{2}, \quad C_1 = -\frac{1}{12}, \quad C_2 = -\tfrac{1}{24}, \\ C_3 &= -\frac{19}{720}, \quad C_4 = -\frac{3}{160}, \quad C_5 = -\tfrac{863}{60480}. \end{aligned} \tag{8.142}$$

Metoda Nordsieck bazată pe algoritmul Adams-Moullin de ordinul p foloseşte vectorul soluţie:

$$z_k = \left[x_k, h x_k', \frac{h^2}{2} x_k'', \ldots, \frac{h^p}{(p+1)!} x_k^{(p+1)}\right]^T. \tag{8.143}$$

Derivata de ordin $(p+2)$ a soluţiei poate fi calculată aproximativ, prin diferenţa divizată, folosind ultima componentă $z^{(p+1)}$ a vectorului Nordsieck la două momente succesive de timp:

$$x^{(p+2)}(\tau) \approx \frac{1}{h}\left[x_k^{(p+1)} - x_{k-1}^{(p+1)}\right] = \frac{(p+1)}{h^{p+2}}\left[z_k^{(p+1)} - z_{k-1}^{(p+1)}\right] = \frac{(p+1)}{h^{p+2}} \nabla z_k^{(p+1)}. \tag{8.144}$$

În urma modificării pasului de timp la valoarea $h' = \alpha h$, eroarea de trunchiere estimată este:

$$\varepsilon_t = C_p \cdot (p+1)! \cdot \alpha^{(p+2)} \cdot \nabla z_k^{(p+1)}. \tag{8.145}$$

Această relaţie permite estimarea coeficientului α de modificare a mărimii pasului de timp, în vederea obţinerii erorii de trunchiere dorite:

$$\alpha \approx \left[\frac{\varepsilon_{\max}}{C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)}}\right]^{\frac{1}{p+2}}. \tag{8.146}$$

Pentru a pondera măririle hazardate de pas, aceste rezultate se ponderează cu o constantă

subunitară. Pentru pasul $p = 0, 1, \ldots, 5$ se propune folosirea unui coeficient optim:

$$\alpha_p = \frac{1}{1,2}\left[\frac{\varepsilon_{\max}}{C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)}}\right]^{\frac{1}{p+2}}. \tag{8.147}$$

Dacă simultan cu modificarea pasului are loc şi o micşorare cu o unitate a ordinului, atunci coeficientul de modificare se calculează cu o relaţie asemănătoare:

$$\alpha_{p-1} = \frac{1}{1,3}\left[\frac{\varepsilon_{\max}}{C_{p-1} \cdot (p+1)! \cdot z_k^{(p+1)}}\right]^{\frac{1}{p+1}} \tag{8.148}$$

cu $p = 1, 2, \ldots, 5$.

Majorarea cu o unitate a ordinului presupune adoptarea unui coeficient de modificare a pasului:

$$\alpha_{p+1} = \frac{1}{1,4}\left[\frac{\varepsilon_{\max}}{C_{p+1} \cdot (p+1)! \cdot \nabla^2 z_k^{(p+1)}}\right]^{\frac{1}{p+3}} \tag{8.149}$$

cu $p = 0, 1, \ldots, 4$. Pentru calculul coeficientului ε_{p+1} este necesară evaluarea diferenţei de ordinul doi:

$$\nabla^2 z_k^{(p+1)} = z_k^{(p+1)} - 2z_{k+1}^{(p+1)} + z_{k+2}^{(p+1)}. \tag{8.150}$$

Ordinul adoptat pe următorul pas se alege corespunzător valorii maxime din cele trei: α_{p-1} (scade ordinul curent), α_p (se păstreză ordinul), α_{p+1} (creşte ordinul curent). Pasul nu este crescut, dacă ordinul curent trebuie menţinut şi $1 < \alpha_p < 1,1$. Pentru a evita modificarea prea frecventă a ordinului sau a mărimii pasului, Gear propune calculul coeficienţilor α, în vederea modificarii pasului, doar în următoarele condiţii:

1. eroarea de trunchiere a depăşit valoarea maxim admisibilă:

$$\varepsilon_t = C_p \cdot (p+1)! \cdot \nabla z_k^{(p+1)} > \varepsilon_{\max},$$

caz în care calculul soluţiei la pasul curent de timp se reia;
2. s-au scurs cel puţin $(p+1)$ paşi de timp de la ultima modificare a ordinului sau a pasului;
3. după 10 paşi de la ultima estimare a parametrilor α care nu a dus la modificarea pasului.

8.8 Integrarea numerică a ecuaţiilor de tip stiff

8.8.1 Ecuaţii diferenţiale de tip stiff

În analiza numerică a sistemelor, inclusiv a circuitelor electrice şi electronice se întâlnesc

situaţii în care metodele de discretizare prezentate anterior nu dau rezultate satisfăcătoare. Spre exemplu, se consideră circuitul simplu din figura 8.23 a, în care cele două bobine sunt parcurse la momentul $t = 0$ de curentul $i_0 = i_1(0) = -i_2(0)$. Se urmăreşte determinarea variaţiei în timp a intensităţii curentului $i(t) = i_1(t) + i_2(t)$, care parcurge comutatorul K, închis la momentul $t = 0$.

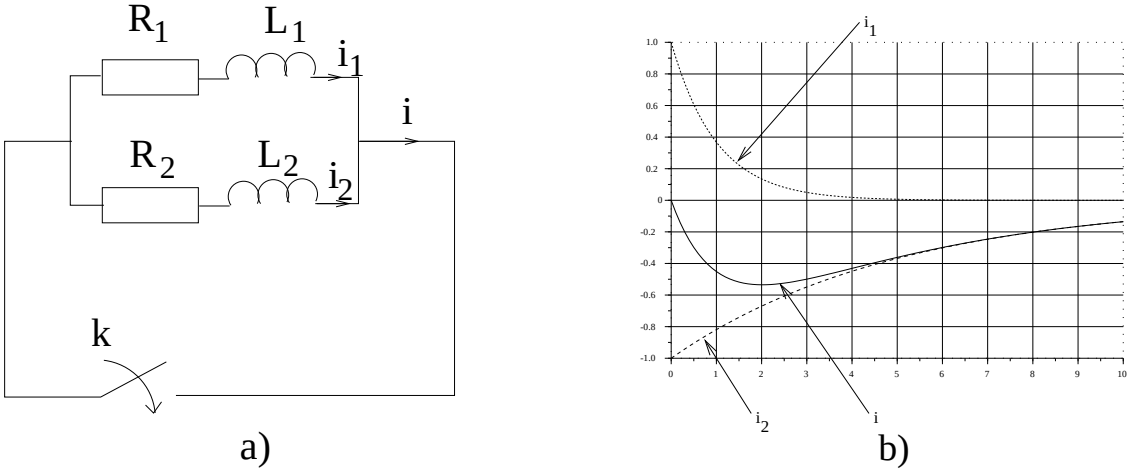


Figura 8.23: Circuit de tip stiff cu două constante de timp: a) Schema circuitului; b) Variaţia în timp a curenţilor i_1, i_2 şi i

Ecuaţiile de stare asociate acestui circuit sunt:

$$\begin{aligned} \frac{di_1}{dt} &= -\frac{R_1}{L_1} i_1; \\ \frac{di_2}{dt} &= -\frac{R_2}{L_2} i_2. \end{aligned} \tag{8.151}$$

Folosind condiţiile iniţiale, rezultă soluţiile:

$$\begin{aligned} i_1(t) &= i_0 \cdot e^{-t/\tau_1}; \\ i_2(t) &= -i_0 \cdot e^{-t/\tau_2}; \\ i(t) &= i_0(e^{-t/\tau_1} - e^{-t/\tau_2}). \end{aligned} \tag{8.152}$$

în care cele două constante de timp au valorile:

$$\tau_1 = \frac{L_1}{R_1}, \ \tau_2 = \frac{L_2}{R_2}. \tag{8.153}$$

Aceste funcţii sunt reprezentate grafic în figura 8.15.b. Dacă se presupune că $\tau_2 \ll \tau_1$, de exemplu : $R_1 = R_2 = 1$ şi $L_1 = 1$ H, $L_2 = 1$ mH, atunci intensitatea curentului $i(t)$ variază rapid în primele milisecunde şi apoi relativ lent în următoarele secunde.

Rezolvarea acestei probleme cu o metodă numerică presupune alegerea unui pas de discretizare mai mic decât cea mai mică dintre constantele de timp, de exemplu:

$h = \tau_2/2 = 0,5$ ms, alegere raţională pentru studiul variaţiei rapide din primele milisecunde. În schimb, pentru timpi mai mari de $5 \div 10$ s, această valoare a pasului este ineficientă, deoarece pentru integrare până la 5 s trebuie efectuaţi circa 1000 de paşi. Dacă se adoptă pe acest interval o valoare a pasului mai mare de $2\tau_2 = 2$ ms, atunci rezultatele vor fi afectate de instabilităţi numerice puternice, datorate componentei rapid variabile, cu constanta de timp τ_2, chiar dacă aceasta este practic "stinsă" pe acest interval.

Acest tip de circuite, caracterizate de constante de timp foarte diferite se numesc circuite de tip *"stiff"*. Metodele clasice de integrare numerică nu se pot aplica la analiza numerică a circuitelor de tip stiff, deoarece ele impun valori ale pasului de timp nejustificat de mici. Dacă în exemplul anterior raportul celor două constante de timp τ_1/τ_2 nu are valoarea 1000, ci 10^6 sau 10^9, atunci timpul de calcul pentru analiza acestui circuit simplu folosind un pas de integrare de ordinul τ_2 devine inacceptabil de mare. Dacă se presupune că, pentru un pas de timp, calculele durează circa 10μs, atunci pentru analiza pe un interval de 1 s, timpul de calcul se ridică la $10^9 \cdot 10\mu\text{s} = 10^4\text{s} \approx 3$ ore.

În exemplul anterior, cele două componente se pot studia separat, deoarece matricea sistemului este de tip diagonal. Se pot da uşor exemple de circuite liniare caracterizate de ecuaţii de stare:

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x} \tag{8.154}$$

de tip stiff. În acest caz valorile proprii ale matricei A au valori ale modulelor foarte diferite. Această caracteristică poate fi generalizată şi pentru cazul circuitelor neliniare.

Un sistem neliniar, caracterizat de ecuaţia de stare:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) \tag{8.155}$$

în care **matricea Jacobian** $J = [\partial\mathbf{f}/\partial\mathbf{x}]$ are valori proprii la care raportul dintre modulul maxim şi cel minim are valori mult mai mari decât unitatea este un *sistem de tip stiff*.

Efectele de tip stiff nu sunt legate numai de valorile proprii ale matricii caracteristice şi se pot obţine şi pe seama excitaţiei circuitului. De exemplu, un circuit cu o singură constantă de timp ca cel din figura (8.24)a, la care sursa variază lent în timp:

$$e(t) = E \cdot e^{-t/\tau_1}, \tag{8.156}$$

cu condiţia iniţială $u(0) = E$, are soluţia:

$$u(t) = \frac{E}{1 - \tau/\tau_1}(e^{-t/\tau_1} - e^{-t/\tau}) + E \cdot e^{-t/\tau}, \quad \tau = RC \tag{8.157}$$

reprezentată grafic în figura (8.24)b. Dacă $\tau \ll \tau_1$, atunci analiza numerică ridică problemele menţionate şi deci, este de tip stiff.

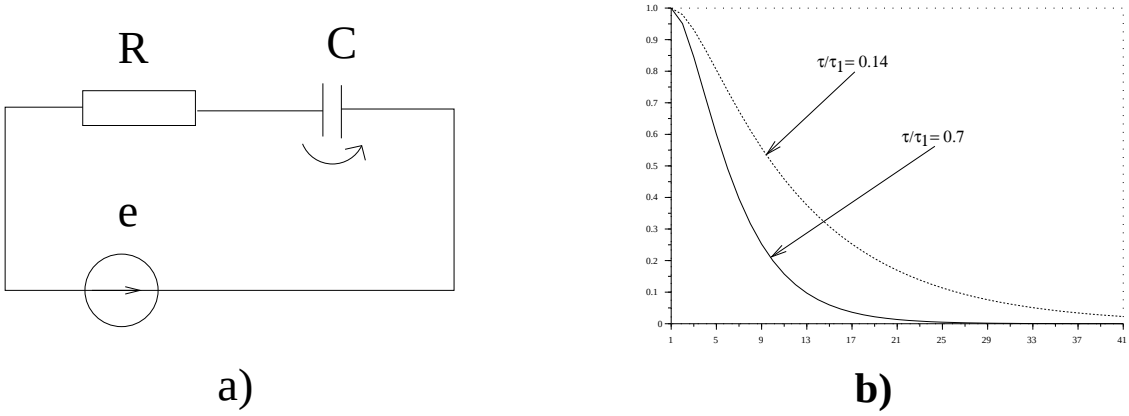


Figura 8.24: Circuit de tip stiff cu o constantă de timp: a) Schema circuitului, cu $e(t) = E \cdot e^{-t/\tau_1}$; b) Variaţia în timp a tensiunii

Un caz special de probleme de tip stiff îl reprezintă **sistemele mixte de ecuaţii diferenţiale şi algebrice**. În acest caz ecuaţiile algebrice pot fi privite ca ecuaţii diferenţiale degenerate:

$$\begin{aligned} x' &= f(x, y, t) \\ \varepsilon y' &= g(x, y, t) \end{aligned} \tag{8.158}$$

în care parametrul $\varepsilon \to 0$. Valorile foarte mici ale parametrului ε corespund unor componente rapid variabile în timp, conform cărora soluţia y a ecuaţiilor algebrice urmăreşte practic instantaneu excitaţiile.

Se constată că nu poate fi formulată o definiţie exactă pentru *sistemele de ecuaţii diferenţiale de tip stiff*, ci doar o caracterizare, prin afirmaţia că un astfel de sistem conţine în soluţia sa cel puţin două componente, una rapid variabilă şi alta lent variabilă faţă de variabila independentă.

În analiza circuitelor electrice şi electronice intervin deseori probleme de tip stiff, mai ales atunci când se introduc capacităţile şi inductivităţile parazite, ale căror valori sunt mult mai mici decât cele utile. Modelele complexe ale dispozitivelor electronice conţin capacităţi sau rezistente ale căror valori pot varia în domenii de $1/10^6$, pentru diferitele regiuni de funcţionare ale componentei. Acestea fac ca un circuit care conţine unul sau mai multe astfel de modele să prezinte efecte stiff.

8.8.2 Stabilitatea numerică a ecuaţiilor stiff

Modul în care a fost anterior definită stabilitatea numerică nu este potrivit pentru abordarea problemelor stiff şi trebuie dată o nouă definiţie destinată acestei probleme speciale. Dintre definiţiile posibile, cea mai larg acceptată este cea datorată lui Gear [26].

Definiţia 8.3 *O metodă numerică, care, aplicată ecuaţiei test:*

$$\frac{dx}{dt} = \lambda x, \tag{8.159}$$

cu $\lambda \in \mathbf{C}$, este absolut stabilă în regiunea R_1 : $Re\,[\lambda h] \leq a \leq 0$ şi este relativ stabilă în regiunea R_2 : $a < Re[\lambda h] \leq b \leq 0, -\theta \leq Im[\lambda h] \leq \theta$, se numeşte numeric stiff stabilă pentru pasul h. Cele două regiuni, R_1 şi R_2, ale planului complex $\mathbf{C}$ sunt reprezentate grafic în figura 8.25.

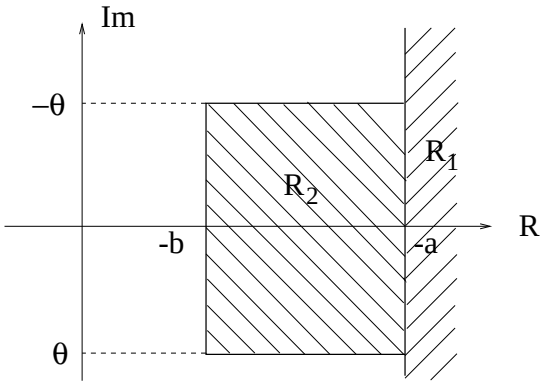


Figura 8.25: Regiuni de stabilitate ale unei metode numerice tip stiff pentru ecuaţia (8.159)

Analizând această definiţie pentru cazul uzual al sistemelor stabile (Re $[\lambda] = -1/\tau < 0$, Im $[\lambda] = \omega$) rezultă că pentru paşi de timp oricât de mari $h \geq a\ \tau$ ($\lambda h \in R_1$), metoda este absolut stabilă, ceea ce face ca eroarea propagată, corespunzătoare componentelor rapid variabile, să nu crească în timpul integrării. Mai mult, pe măsură ce valoarea acestor componente tinde către zero şi eroarea asociată lor se anulează.

Pentru valori mici ale pasului de integrare $h < \min[a\tau,\ \theta/\omega]$, care corespund condiţiei $\lambda h \in R_2$, metoda este relativ stabilă, ceea ce asigură acurateţe numerică soluţiei obţinute.

Faptul că R_2 se extinde şi în semiplanul $\lambda h > 0$, permite abordarea şi a problemelor instabile (Re$[\lambda] = \alpha > 0$, Im $[\lambda] = \omega$), caz în care pasul de timp trebuie să fie suficient de mic $h < \min\,[b/\alpha,\ \theta/\omega]$ pentru a obţine soluţii cu acurateţe acceptabilă şi neafectate grav de instabilităţi numerice.

Analizând domeniile de stabilitate numerică absolută ale metodelor anterior definite se constată că regiunea R_1 este conţinută doar în domeniile metodelor implicite Adams-Moulton cu $p = 0$ şi $p = 1$. Doar metoda trapezelor şi metoda Euler implicită satisfac condiţiile de stabilitate numerică stiff, pentru care $a = 0$.

Dacă se abordează problema din figura 8.16 folosind metoda Euler explicită, atunci soluţia numerică prezintă instabilităţi inacceptabile (figura 8.26a), pe când cu metoda Euler implicită se obţin chiar la pas mare de timp soluţii numerice fără instabilităţi (figura 8.26b).

Dalquist a demonstrat că nu există metode multipas de ordin superior metodei trapezelor care să fie absolut stabile în semiplanul λh stâng (stiff numeric stabile cu $a = 0$).

Gear a găsit o clasă de metode numerice multipas de ordin superior, care sunt stiff numeric stabile pentru $a < 0$, pornind de la studiul comportării polinomului caracteristic (8.89) pentru $\lambda h \to -\infty$.

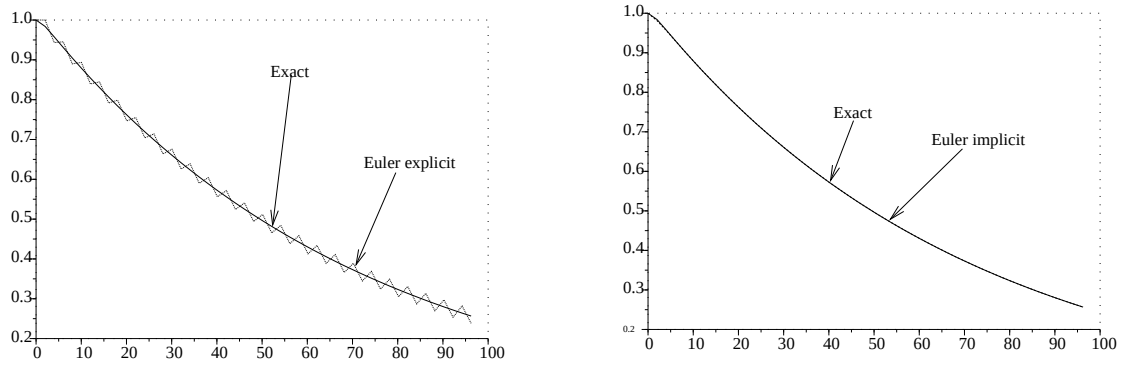


Figura 8.26: Soluţiile unei ecuaţii stiff prin metodele Euler explicită (a) şi implicită (b)

Rădăcinile polinomului caracteristic $P(z)$ satisfac egalitatea:

$$\lambda h = -\frac{A(z)}{B(z)}, \tag{8.160}$$

iar dacă $\lambda h \to -\infty$, rezultă că:

$$gradB(z) \geq gradA(z) \tag{8.161}$$

deoarece în caz contrar punctul de la infinit este rădăcină a polinomului caracteristic şi metoda îşi pierde stabilitatea numerică absolută. Din inegalitatea (8.161), rezultă că metodele multipas stiff sunt în mod necesar metode implicite ($b_0 \neq 0$). Nu pot exista metode explicite care să fie numeric stiff stabile.

Împărţind relaţia (8.89) la λh, rezultă că rădăcinile polinomului caracteristic satisfac:

$$\frac{P(z)}{\lambda h} = \frac{1}{\lambda h} A(z) + B(z) = 0, \tag{8.162}$$

sau la limită, pentru $\lambda h \to -\infty$, rezultă $B(z) = 0$, sau dezvoltat:

$$b_0 z^p + b_1 z^{p-1} + ... + b_p = 0.$$

Pentru ca rădăcinile parazite să aibă modul minim la un pas oricât de mare, se alege:

$$b_1 = b_2 = ... = b_p = 0, \tag{8.163}$$

ceea ce corespunde polinomului $B(z) = b_0 = z^p$ care are toate cele p rădăcini multiple şi egale cu 0. Pe baza continuităţii rădăcinilor faţă de coeficienţii unui polinom, este de aşteptat ca şi pentru alte valori ale lui λh rădăcinile parazite să aibă un modul cât mai mic. Cu aceste alegeri, rezultă relaţia recurentă de integrare numerică cu metoda Gear:

$$x_k = b_0 h \cdot f(x_k, t_k) + \sum_{i=1}^{p} a_i x_{k-i}. \tag{8.164}$$

Coeficienţii b_0 şi a_i se determină pe baza condiţiilor de consistentă (8.74), (8.75):

$$\begin{aligned} &\sum_{i=0}^{p} a_i = 0; \\ &\sum_{i=1}^{p} (1-i)^j a_i + j b_0 = 0, \quad j = 1, 2, ..., p, \end{aligned} \tag{8.165}$$

în care, dacă se consideră $a_0 = -1$, rezultă un sistem de $(p+1)$ ecuaţii:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 0 \\ 0 & -1 & -2 & \dots & (p-1) & 1 \\ 0 & 1 & 4 & \dots & (p-1)^2 & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & (-1)^p & (-2)^p & \dots & (1-p)^p & p \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ b_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} \tag{8.166}$$

cu $(p+1)$ necunoscute: $b_0, a_1, a_2, ..., a_p$.

Pentru cazul $p = 1$, se obţine:

$$\begin{aligned} a_1 &= 1 \\ b_0 &= 1, \end{aligned}$$

ceea ce corespunde la $x_k = hf(x_k, t_k) + x_{k-1}$, care este chiar relaţia Euler implicită.

Dacă se consideră $p = 2$, se obţine sistemul

$$\begin{aligned} &a_1 + a_2 = 1; \\ &-a_2 + b_0 = 1; \\ &a_2 + 2b_0 = 1, \end{aligned}$$

care are soluţiile: $a_1 = 4/3, a_2 = -1/3$ şi $b_0 = 2/3$, ceea ce corespunde relaţiei Gear de ordinul doi:

$$x_k = \frac{2}{3} h \cdot f(x_k, t_k) + \frac{1}{3}(4x_{k-1} - x_{k-2}). \tag{8.167}$$

În mod asemănător se determină coeficienţii metodelor Gear de ordin superior, coeficienţi ale căror valori sunt prezentate în tabelul 8.6:

Pentru determinarea domeniilor de stabilitate absolută se consideră mulţimea punctelor $z \in \mathbf{C}$ care au modul unitar: $z = e^{j\varphi}$ cu $j = \sqrt{-1}$. Dacă aceste puncte sunt rădăcini ale polinomului caracteristic, atunci locul geometric al punctelor:

$$s = \lambda h = -\frac{A(z)}{B(z)} = -\frac{A(e^{j\varphi})}{B(e^{j\varphi})} \tag{8.168}$$

reprezintă frontiera domeniului de stabilitate absolută din planul complex λh.

Pentru metodele Gear aceste curbe au ecuaţiile:

$$s = -\frac{1}{b_0} \sum_{i=0}^{p} a_i e^{-ji\varphi}, \quad \in [0,\ 2\pi]. \tag{8.169}$$

Tabela 8.6: Coeficienţii metodei Gear

p	ν	νb_0	νa_1	νa_2	νa_3	νa_4	νa_5	νa_6
1	1	1	1					
2	3	2	4	-1				
3	11	6	18	-9	2			
4	25	12	48	-36	16	-3		
5	137	60	300	-300	200	-75	12	
6	147	60	360	-450	400	-275	72	-10

În cazul $p = 1$, ecuaţia parametrică:

$$s(\varphi) = 1 - e^{j\varphi} \tag{8.170}$$

evidenţiază un cerc cu centrul în $s_0 = 1$ şi raza unitate. Punctele exterioare cercului determină rădăcini cu modul subunitar, de exemplu pentru $\lambda h = -1$,

$$\begin{aligned} &P(z) = A(z) + \lambda h B(z) = -z + 1 + \lambda h z = z(\lambda h - 1) + 1, \\ &z = -\tfrac{1}{\lambda h - 1} = \tfrac{1}{2}. \end{aligned} \tag{8.171}$$

Rezultă că întregul plan complex minus discul circular menţionat reprezintă domeniul de stabilitate absolută.

În cazul $p = 2$, ecuaţia frontierei domeniului de stabilitate este:

$$s(\varphi) = \frac{3}{2} - 2e^{j\varphi} + \frac{1}{2}e^{2j\varphi}, \tag{8.172}$$

care reprezintă o curbă închisă simetrică faţă de axa imaginară, ce trece prin origine şi punctul $\lambda h = 4$. Si de această dată domeniul de stabilitate este exterior acestei curbe deoarece polinomul caracteristic:

$$P(z) = A(z) + \lambda h B(z) = -z^2 + \frac{4}{3}z - \frac{1}{3} + \lambda h \frac{2}{3}z^2$$

are rădăcinile:

$$\begin{aligned} z_1 &= \frac{2 + \sqrt{1 + 2\lambda h}}{3 - 2\lambda h}; \\ z_2 &= \frac{2 - \sqrt{1 + 2\lambda h}}{3 - 2\lambda h}, \end{aligned} \tag{8.173}$$

cu modul subunitar, de exemplu pentru $\lambda h = -1/2$, iar în cerc $z_1 = z_2 = 1/2$.

Domeniile de stabilitate absolută ale metodelor Gear de diferite ordine sunt prezentate în figura 8.27. Se constată că limita domeniului de stabilitate absolută are valorile $a = 0$, pentru ordinele unu şi doi, iar pentru ordinele de la trei până la şase, aceste limite au

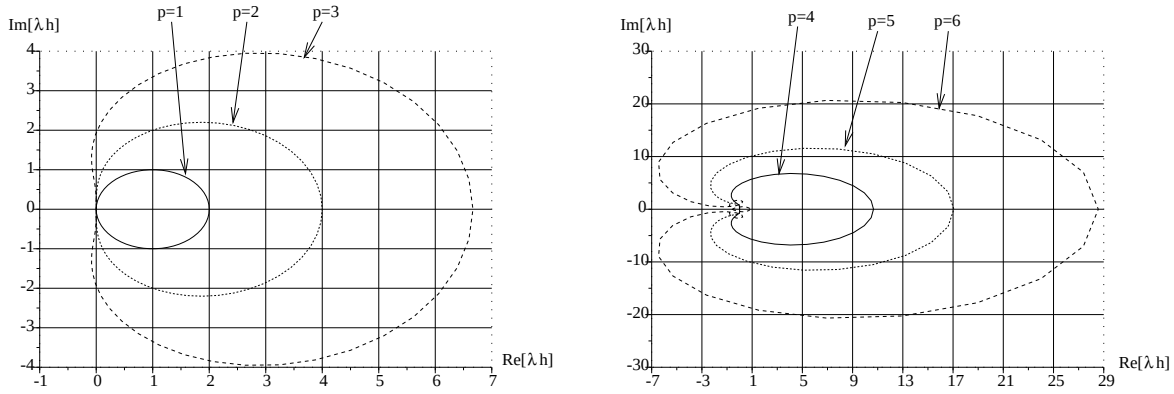


Figura 8.27: Domenii de stabilitate numerică pentru metodele Gear de diferite ordine p

valorile $a = -0.1, -0.7, -2.4$ şi respectiv $a = -6.1$. Pentru ordine superioare lui 6 metodele Gear nu mai sunt numeric stiff stabile.

Dacă se reprezintă modulele celor două rădăcini $|z_1|$ şi $|z_2|$ în funcţie de parametrul λh presupus real se obţine graficul din figura 8.28, care evidenţiază caracterul numeric absolut stabil al metodelor Gear de ordinul doi, pentru $\lambda h \leq 0$ şi numeric relativ stabil, pentru $0 < \lambda h < 3/2$.

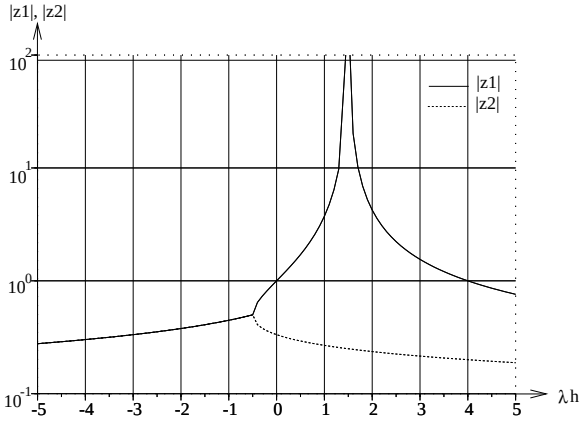


Figura 8.28: Dependenţa de λh a rădăcinilor polinomului caracteristic pentru metoda Gear de ordinul doi

În mod asemănător se demonstrează că pentru metodele Gear de ordine cuprinse între 1 şi 6 există o regiune R_2 în vecinătatea originii, pentru care algoritmii sunt numeric relativ stabili. Rezultă că toate aceste metode se bucură de proprietatea stabilităţii numerice de tip stiff.

8.8.3 Algoritmul Gear pentru rezolvarea ecuaţiilor stiff

Deoarece metodele Gear sunt metode multipas de tip implicit, la fiecare pas de integrare este necesară rezolvarea ecuaţiei:

$$x_k = hb_0 \cdot f(x_k, t_k) + \sum_{i=1}^{p} a_i x_{k-i}. \tag{8.174}$$

Dacă pasul de integrare este suficient de mic:

$$h < \frac{1}{b_0 L_k}, \tag{8.175}$$

unde L_k este constanta Lipschitz a funcţiei $f(x,t)$ pentru $t = t_k$, atunci iteraţia simplă:

$$x_k^{(j+1)} = hb_0 \cdot f(x_k^{(j)}, t_k) + \sum_{i=1}^{p} a_i x_{k-i}, \tag{8.176}$$

este convergentă.

Deoarece în cazul ecuaţiilor stiff pasul de timp poate fi mult mai mare decât constanta de timp a componentelor rapide, inegalitatea (8.175) nu este în general satisfăcută. Din acest motiv rezolvarea ecuaţiei (8.174) prin metoda iteraţiei simple, bazată pe relaţia (8.176) nu este folosită, fiind neconvergentă.

Ecuaţiei neliniară implicită adusă sub forma:

$$x_k - hb_0 \cdot f(x_k, t_k) - \sum_{i=0}^{p} a_i x_{k-i} = 0 \tag{8.177}$$

se poate rezolva iterativ cu metoda Newton-Raphson:

$$x_k^{(j+1)} = x_k^{(j)} - [I - hb_0 \cdot J_f(x_k^{(j)}, t_k)]^{-1} \cdot \left\{ x_k^{(j)} - hb_0 \cdot f(x_k^{(j)}, t_k) - \sum_{i=1}^{p} a_i x_{k-1} \right\}, \tag{8.178}$$

în care I este matricea unitate de dimensiune $n \times n$, iar:

$$J_f(x_k^{(j)}, t_k) = \left. \frac{\partial f(x,t)}{\partial x} \right|_{x = x_k^{(j)}, t = t_k} \tag{8.179}$$

este matricea Jacobian a funcţiei $f(x,t)$, matrice de dimensiune $n \times n$.

Deoarece termenul care conţine coeficienţii $a_1, a_2, ..., a_p$ este constant pe parcursul iteraţiilor, el poate fi eliminat. Dacă se scad relaţiile corespunzătoare a două iteraţii succesive:

$$[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] = -x_k^{(j)} + hb_0 \cdot f(x_k^{(j)}, t_k) + \sum_{i=1}^{p} a_i x_{k-i}$$

$$[I - b_0 h \cdot J_f(x_k^{(j-1)}, t_k)] \cdot [x_k^{(j)} - x_k^{(j-1)}] = -x_k^{(j-1)} + hb_0 \cdot f(x_k^{(j-1)}, t_k) + \sum_{i=1}^{p} a_i x_{k-i}$$

rezultă:

$$\begin{aligned}&[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] = h b_0 [f(x_k^{(j)}, t_k) - f(x_k^{(j-1)}, t_k)] - \\ &- h b_0 \cdot J_f(x_k^{(j-1)}, t_k) \cdot [x_k^{(j)} - x_k^{(j-1)}],\end{aligned} \tag{8.180}$$

în care dacă se notează:

$$d_k^{(j)} = h \cdot f(x_k^{(j-1)}, t_k) + h \cdot J_f(x_k^{(j-1)}, t_k)[x_k^{(j)} - x_k^{(j-1)}], \tag{8.181}$$

rezultă:

$$b_0[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [x_k^{(j+1)} - x_k^{(j)}] = h x_k'^{(j)} - d_k^{(j)}, \tag{8.182}$$

deoarece

$$x_k'^{(j)} = f(x_k^{(j)}, t_k).$$

Scăzând două valori succesive pentru $d_k^{(j)}$ şi $d_k^{(j+1)}$ se constată că şi acestea satisfac o relaţie de recurenţă asemănătoare:

$$[I - b_0 h \cdot J_f(x_k^{(j)}, t_k)] \cdot [d_k^{(j+1)} - d_k^{(j)}] = h x_k'^{(j)} - d_k^{(j)}. \tag{8.183}$$

Aplicarea acestor relaţii iterative permite determinarea limitelor şirurilor:

$$\begin{aligned}&x_k^{(1)}, x_k^{(2)}, ..., x_k^{(j)}, ... \to x_k; \\ &d_k^{(1)}, d_k^{(2)}, ..., d_k^{(j)}, ... \to d_k = h x_k',\end{aligned} \tag{8.184}$$

pornind de la iniţializarea:

$$d_k^{(0)} = \frac{1}{b_0} \left[x_k^{(0)} - \sum_{i=1}^{p} a_i x_{k-i} \right], \tag{8.185}$$

obţinută prin identificarea relaţiilor (8.176) şi (8.181) pentru $j = 0$.

Pentru ca schimbarea pasului de integrare să se facă uşor, metoda Gear se implementează în reprezentarea Nordsieck (121):

$$z_k = \left[x_k,\ h x_k',\ \frac{h^2}{2} x_k'', ..., \frac{h^p}{p!} x_k^{(p)} \right]^T.$$

În acest caz, relaţia de predicţie are reprezentarea matriceală:

$$z_k^{(0)} = Z z_{k-1}, \tag{8.186}$$

în care matricea Z este de tip triunghi Pascal, iar relaţia de corecţie este:

$$z_k^{(j+1)} = z_k^{(j)} + G(z_k^{(j)}) \cdot c_z. \tag{8.187}$$

Se constată că $d_k^{(j)}$ fiind o aproximare a lui $x_k'^{(j)}$, poate fi preluat din componenta secundă a vectorului Nordsieck.

Forma matriceală a corectorului poate fi aplicată şi în cazul iteraţiei Newton-Raphson, dar cu observaţia că funcţia de corecţie G are în acest caz un caracter matriceal:

$$G(z_k^{(j)}) = [I - hb_0 \cdot J_f(x_k^{(j)}, t_k)]^{-1} \cdot [hx_k'^{(j)} - d_k^{(j)}]. \tag{8.188}$$

Pentru determinarea vectorului de corecţie c_z trebuie determinată matricea de transformare T. Se consideră spre exemplificare cazul $p = 2$. Fie polinomul de grad doi:

$$x(t) = a_0 + a_1 t + a_2 t^2$$

cu derivatele:

$$x' = a_1 + 2a_2 t; \quad x'' = 2a_2,$$

care pe o reţea regulată $t_k = 0, t_{k-1} = -h$ are vectorul Gear canonic:

$$y_k = [x_k,\ x_{k-1},\ hx_k']^T = [a_0,\ a_0 - 2a,\ h + 2a_2 h^2,\ ha_1]^T,$$

iar vectorul Nordsieck:

$$z_k = \left[x_k, hx_k', \frac{h^2}{2}x_k''\right]^t = [a_0, ha_1, h^2 a_2]^t.$$

Transformarea de echivalenţă între cei doi vectori este:

$$y_k = \begin{bmatrix} x_k \\ x_{k-1} \\ hx_k' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -2 & 2 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ ha_1 \\ h^2 a_2 \end{bmatrix} = T^{-1} z_k.$$

Prin inversare rezultă:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -2 & 2 \\ 0 & 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \tag{8.189}$$

matrice care permite calculul vectorului de corecţie:

$$c_z = Tc_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2/3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 3/3 \\ 1/3 \end{bmatrix}, \tag{8.190}$$

pornind de la $c_y = [b_0 \quad 0 \quad 1]^T$.

Valorile componentelor vectorilor de corecţie c_z, pentru diferite ordine, sunt date în tabelul 8.7:

Tabela 8.7: Componentele vectoriale de corecţie

p	i =	1	2	3	4	5	6	7
2	3 c_z(i)	2	3	1				
3	11 c_z(i)	6	11	6	1			
4	50 c_z(i)	24	50	35	10	1		
5	274 c_z(i)	170	274	225	85	15	1	
6	1764 c_z(i)	720	1724	1624	735	175	21	1

Comparând cele două variante ale algoritmilor predictor-corector în reprezentarea Nordsieck se constată o mare asemănare între ele. Deosebirea de principiu constă în faptul că la metoda Gear, fiecare iteraţie a corectorului presupune calculul matricii Jacobian J_f şi rezolvarea unui sistem de ecuaţii de dimensiunea acestei matrici, pe când la metodele de tip Adams este necesară doar evaluarea funcţiilor $f(x,t)$. Efortul de calcul suplimentar este "tributul" plătit pentru a asigura stabilitatea numerică a ecuaţiilor stiff şi pentru a putea avansa cu un pas de integrare oricât de mare după ce componentele rapid variabile s-au "stins". Se constată că matricea $I - hb_0 J_f$ nu trebuie neapărat inversată, deoarece în majoritatea cazurilor această matrice nu variază prea mult de la o iteraţie la alta, motiv pentru care ea poate fi presupusă constantă şi inversată o singură dată.

Tehnicile aplicate pentru controlul automat al ordinului şi al pasului de integrare, prezentate în paragraful anterior pot fi aplicate şi în acest caz.

În metoda Gear, eroarea de trunchiere satisface egalitatea:

$$\varepsilon = \left[C_p \cdot \hat{x}^{(p+1)}(\tau)\right] h^{p+1} = O(h^{p+1}),$$

în care coeficienţii au valorile:

$$c_1 = -\tfrac{1}{2}, \quad c_2 = -\tfrac{2}{9}, \quad c_3 = -\tfrac{3}{22},$$

$$c_4 = -\tfrac{12}{125}, \quad c_5 = -\tfrac{10}{137}, \quad c_6 = -\tfrac{60}{1029}.$$

În consecinţă, în reprezentarea Nordsieck, eroarea de trunchiere poate fi calculată cu o relaţie de forma (8.139), dar această eroare se poate calcula eficient, din punctul de vedere al programatorului, cumulând corecţiile ce trebuie aplicate soluţiei $x_k^{(j)}$, pe parcursul iteraţiilor, respectiv pentru M suficient de mare:

$$c_z = \sum_{j=0}^{M} G(z_k^{(j)}) = z_k^{(M)} - z_k^{(0)} = z_k - z_k^{(0)},$$

iar dacă se reţine ultima componentă:

$$c_{z(p+1)} \sum_{j=0}^{M} G(z_k^{(j)}) = z_{k(p)} - z_{k(p)}^{(0)} = z_{k(p)} - z_{k-1(p)} = \frac{hp}{p!}(x_k^{(p)} - x_{k-1}^{(p)}) \cong \frac{h^{p+1}}{p!} x_k^{(p+1)},$$

de unde rezultă că:

$$\varepsilon = c_p \cdot p! \cdot c_{z(p+1)} \sum_{j=0}^{M} G(z_k^{(j)}).$$

8.8.4 Alte metode numeric stiff stabile

În afara metodelor de tip Gear şi a metodei trapezelor, mai există şi alte metode de discretizare numeric stiff stabile. Dintre acestea menţionăm varianta implicită a metodelor Runge - Kutta. Varianta cea mai simplă a acestor metode, numită *a punctului median* se bazează pe relaţiile:

$$\begin{aligned} g_1 &= h \cdot f\left(x_k + \tfrac{1}{2}g_1, t_k + \tfrac{1}{2}h\right) \\ x_{k+1} &= x_k + g_1. \end{aligned} \tag{8.191}$$

Se constată că incrementul g_1, care trebuie aplicat soluţiei pentru a avansa cu un pas de timp, se determină prin rezolvarea unei ecuaţii neliniare implicite, corespunzătoare momentului de timp median $t_k + h/2$.

Metoda Runge - Kutta implicită cu două "evaluări" are forma generală:

$$x_{k+1} = x_k + ha_1 g_1 + a_2 g_2, \quad \text{cu}$$

$$\begin{aligned} g_1 &= f(x_k + b_1 g_1 + b_2 g_2, t_k + c_1 h); \\ g_2 &= f(x_k + b_3 g_3 + b_4 g_4, t_k + c_2 h). \end{aligned} \tag{8.192}$$

Pentru următoarea alegere a coeficienţilor:

$$\begin{aligned} a_1 &= a_2 = 1/2; & b_2 &= 1/4 - \sqrt{3}/6; \\ b_1 &= b_4 = 1/4; & b_3 &= 1/4 + \sqrt{3}/66; \\ c_1 &= 1/2 - \sqrt{3}/6; & c_2 &= 1/2 + \sqrt{3}/6, \end{aligned}$$

metoda asigură o eroare globală de trunchiere de tipul $O(h4)$, deci este de ordinul patru. Deoarece la fiecare moment de timp trebuie rezolvat un sistem de ecuaţii neliniare de dimensiune dublă faţă de cel din varianta anterioară, efortul de calcul asociat acestei metode este foarte mare.

Pentru a evita rezolvarea ecuaţiilor neliniare, Rosenbrock propune o categorie de metode bazate pe calculul matricei Jacobian $J_f = \frac{\partial f}{\partial x}$. Varianta Calahan, cu două etape a acestei metode:

$$\begin{aligned} g_1 &= h \cdot f(x_k, t_k) + hb_1 J_{f_k} g_1 \\ g_2 &= h \cdot f(x_k + b_2 g_1, t_k + c_1 h) + hb_1 J_{f_k} g_2 \\ x_{k+1} &= x_k + a_1 g_1 + a_2 g_2 \end{aligned}$$

cu $a_1 = 3/4,\ a_2 = 1/4,\ b_1 = (1 + 1/\sqrt{3})/2,\ b_2 = -2\sqrt{3}$ asigură o eroare globală de tipul $O(h3)$, deci este de ordinul 3.

Toate aceste metode conţin semiplanul stâng Re $[\lambda h] < 0$ în domeniul de stabilitate numerică absolută, deci sunt numeric stiff stabile cu parametrul $a = 0$ [26].

O altă categorie de metode, cunoscută sub numele de *metode cu diferenţiere regresivă*, se obţin, în principiu, prin rescrierea relaţiei Gear (8.164) sub forma:

$$x'_k = \frac{1}{hb_0}\left[a_0 x_k - \sum_{i=1}^{p} a_i x_{k-i}\right]. \tag{8.193}$$

Constatând că relaţia (8.193) este o relaţie de derivare numerică cu aproximare polinomială regresivă, pentru a permite manipularea pasului variabil, Brayton propune folosirea relaţiei de interpolare cu paşi inegali, ca relaţie de corecţie:

$$x'_k = \frac{1}{h}\sum_{i=0}^{p} a_i x_{k-i}, \tag{8.194}$$

în care $h = t_k - t_{k-1}$:

$$a_i = \frac{t_k - t_{k-1}}{t_k - t_{k-i}} \prod_{\substack{j=1 \\ j \neq i}}^{p} \frac{t_k - t_{k-i}}{t_{k-i} - t_{k-j}}, \quad j = 1...p \quad \text{şi} \quad \sum_{j=0}^{p} a_j = 0.$$

Pentru predictor se propune o relaţie de acelaşi ordin:

$$x_k^{(0)} = \sum_{i=0}^{p+1} c_i x_{k-i} \text{cu} \quad a_i = \frac{t_k - t_{k-1}}{t_k - t_{k-i}} \prod_{\substack{j=1 \\ j \neq i}}^{p} \frac{t_k - t_{k-i}}{t_{k-i} - t_{k-j}}, \quad j = 1, ..., p.$$

Eroarea locală de trunchiere este dată de:

$$\varepsilon_k = \frac{h}{t_k - t_{k-p-1}}(x_k - x_k^{(0)}). \tag{8.195}$$

Printr-o transformare de echivalenţă, derivata se poate exprima în funcţie de diferenţele finite de ordinul întâi, aşa cum propune Branin:

$$x'_k = \sum_{i=0}^{p} \frac{a_i}{h_i} \nabla x_{k-i}, \tag{8.196}$$

în care: $h_i = t_k - t_{k-p}$, $a_i = \sum_{j=0}^{i} p_i$, iar coeficienţii p_j sunt daţi de: $\sum_{j=0}^{p} p_j = 0$, $\sum_{j=1}^{p} d_j p_j = -1$, în care $d_j = h_j/h_1$.

În acest caz eroarea locală de trunchiere este:

$$\varepsilon = \frac{h^{p+1} x_k^{(p+1)}}{p+1} \prod_{j=1}^{p} \frac{d_j}{j}.$$

O reprezentare, asemănătoare celei Nordsieck, dar folosind diferenţe finite, este propusă de Ruhner-Petersen:

$$x'_{k+1} = \frac{1}{h} \sum_{i=0}^{p-1} a_i b_i \nabla^i x_k. \tag{8.197}$$

unde

$$a_i = \sum_{j=p-1}^{i} \frac{h_1}{t_{k+1} - t_{k-j}}, \quad b_i = \prod_{j=1}^{i} \frac{t_{k+1} - t_{k+1-j}}{t_k - t_{k-j}}, \quad b_0 = 1.$$

Relaţia de predicţie corespunzătoare este:

$$x_{k+1}^{(0)} = \sum_{i=0}^{p} b_i \nabla^i x_k. \tag{8.198}$$

Aceste trei metode fiind echivalente, au aceeaşi eroare de trunchiere şi sunt numeric stiff stabile pentru $1 \leq p \leq 6$. Fiind metode implicite, soluţia se determină prin iteraţii Newton- Raphson, iar modificarea pasului şi a ordinului se poate face prin algoritmul Gear, deoarece sunt cunoscute estimări ale erorii.

Faţă de metoda Gear, aceste metode oferă o stabilitate numerică mai bună, în cazul schimbării frecvente a mărimii pasului, dar ele impun calcularea coeficienţilor a_i la fiecare pas de integrare.

8.9 Analiza numerică a circuitelor electrice în regim tranzitoriu

Prin analiza numerică a circuitelor electrice în regim tranzitoriu se urmăreşte determinarea variaţiei în timp a curenţilor prin laturile unui circuit cu rezistoare, bobine şi condensatoare, alimentate de la surse de tensiune sau curent variabile în timp.

Pentru rezolvarea numerică a ecuaţiilor diferenţiale asociate unui astfel de circuit, intervalul de timp al analizei $[t_0, t_{\max}]$, este împărţit într-o reţea de discretizare. În fiecare nod al reţelei problema se reduce la rezolvarea unui circuit electric rezistiv, prin înlocuirea elementelor acumulatoare de energie (bobine şi condensatoare) din circuitul iniţial, cu modele echivalente. Aceste modele, alcătuite din rezistenţe şi surse, rezultă din discretizarea ecuaţiei diferenţiale ce caracterizează funcţionarea elementului respectiv şi au un caracter rezistiv. Dacă circuitul electric este neliniar, atunci modelul discretizat reprezintă un circuit rezistiv neliniar, pentru analiza căruia se aplică de obicei metode iterative.

8.9.1 Principiul metodei

Se consideră un circuit electric cu L laturi. Fiecare latură poate conţine un element pasiv (R, L, sau C) şi în serie o sursă. Pentru descrierea circuitului sunt necesare următoarele informaţii:

a. topologia circuitului electric - în acest scop fiecărei laturi de circuit i se asociază un nod iniţial ni şi un nod final nf, sensul de referinţă pentru curent, tensiunea la borne fiind de la nodul iniţial către nodul final;

b. tipul elementului pasiv şi parametrii caracteristici ai elementelor pasive din fiecare latură R, L, sau C (pentru care nu pot avea valori nule);

c. parametrii caracteristici ai surselor E (pentru care pot avea şi valori nule);

d. condiţiile iniţiale în cazul acumulatoarelor de energie (tensiunile iniţiale pe condensatoare u_{C_0}, respectiv curenţii iniţiali prin bobine i_{L_0}).

Intervalul de timp $[t_0, t_{\max}]$ este împărţit într-o reţea de discretizare uniformă cu pasul h şi n noduri t_k:

$$t_0 = 0;\ t_1 = h;\ t_2 = 2h;\ \ldots t_n = nh = t_{\max}. \tag{8.199}$$

Se consideră o latură ce conţine o bobină de inductivitate L. La momentul t_k, curentul prin bobină este i_k, iar tensiunea la bornele sale este:

$$u_k = L\frac{\mathrm{d}i}{\mathrm{d}t}. \tag{8.200}$$

Prin discretizare cu metoda Euler implicită, expresia (8.200) devine:

$$u_k = L\frac{i_k - i_{k-1}}{t_k - t_{k-1}}. \tag{8.201}$$

de unde:

$$u_k = i_k\frac{L}{h} - i_{k-1}\frac{L}{h}. \tag{8.202}$$

Ecuaţia (8.202) reprezintă tensiunea la bornele unei laturi ce conţine un rezistor de rezistenţă $R_k = L/h$ şi o sursă de tensiune de valoare

$$E_k = R_k i_{k-1},$$

orientată după sensul de referinţă al laturii.

Folosind acelaşi tip de discretizare, curentul printr-un condensator, la momentul t_k este:

$$i_k = C\frac{u_k - u_{k-1}}{h}, \tag{8.203}$$

ceea ce este echivalent cu o conductanţă $G_k = C/h$ în paralel cu o sursă de curent $J_k = G_k u_{k-1}$. Sursa echivalentă de tensiune a acestei grupări paralel are tensiunea electromotoare $E_k = -u_{k-1}$ şi o rezistenţă internă $R_k = 1/G_k = h/C$. Prin urmare, pentru un condensator:

$$u_k = i_k \frac{h}{C} + u_{k-1}. \tag{8.204}$$

Circuitele R, E echivalente la fiecare pas cu elementele acumulatoare de energie L, C se numesc circuite discretizate.

Pornind de la condiţiile iniţiale i_0, respectiv u_0 se pot determina succesiv potenţialele nodurilor, v_k şi curenţii prin laturi i_k la orice moment $t_k (k = 1, 2, 3, ...)$ prin rezolvarea unui circuit rezistiv (R,E), constituit din circuite discretizate.

Cea mai simplă metodă de rezolvare a acestui circuit este metoda nodală care presupune rezolvarea următorului sistem de ecuaţii liniare:

$$Gv = i_s, \tag{8.205}$$

unde G este matricea conductanţelor, v este vectorul coloană al potenţialelor nodurilor, iar i_s este vectorul coloană al curenţilor de scurtcircuit corespunzători fiecărui nod.

8.9.2 Pseudocodul metodei modelului discretizat

Următorul pseudocod descrie metoda de analiză a circuitelor electrice liniare de tip R, L, C, excitate cu surse de tensiune treaptă în regim tranzitoriu.

```
; Descriere circuit
citeşte L                ; număr de laturi
citeşte N                ; număr de noduri
pentru k = 1,L
        citeşte ni(k), nf(k)     ; noduri iniţial, final ale
                                 ; laturii k
        citeşte tip(k)           ; tipul laturii k (R,L sau C)
        citeşte par(k)           ; parametrul elementului pasiv
                                 ; (R(K), L(k) sau C(k))
        citeşte e(k)             ; t.e.m.  din latura k
        dacă tip(k) ≠ R
                citeşte ci(k)            ; condiţia iniţială din
                                         ; latura k
citeşte tmax             ; timpul maxim al simulării
citeşte h                ; pasul de simulare
t = 0
; Parcurge timpul de simulare
cât timp t ≤ tmax
```

```
t = t + h
; Generează circuit discretizat
pentru k = 1,L
        dacă tip(k) = R atunci
                R(k) = par(k)
                E(k) = e(k)
        altfel dacă tip(k) = L atunci
                R(k) = par(k)/h
                E(k) = e(k) + R(k) · ci(k)
        altfel
                R(k) = h/par(k)
                E(k) = e(k) - ci(k)
; Rezolvă circuit rezistiv prin metoda nodală şi
; determină potenţialele V
nodal ( L, N, ni, nf, R, E, V )
; Actualizează condiţii iniţiale şi afişează soluţia
pentru k = 1,L
        u = V(ni(k)) - V(nf(k))
        i = (E(k) + u)/R(k)
        dacă tip(k) = L atunci
                ci(k) = i
        altfel ci(k) = u - e(k)
        scrie k, u, i
```

8.9.3 Analiza algoritmului

Efort de calcul

Efortul de calcul este determinat în cea mai mare măsură de rezolvarea, la fiecare pas, a circuitului rezistiv prin metodă nodală. Funcţia "`nodal`" generează matricele G şi i_s, cu un efort de calcul de ordinul $O(6L)$ şi rezolvă sistemul liniar (8.205) cu un efort de calcul de ordinul $O(N^3/3)$. Dacă se notează $n = t_{\max}/h$ numărul paşilor de timp, efortul de calcul global este de ordinul $O(n \cdot N^3/3)$ şi constă în principal în rezolvarea de n ori a unui sistem liniar cu o dimensiune egală cu numărul nodurilor circuitului.

O reducere substanţială a timpului de calcul se poate obţine, dacă se observă că la pasul h constant, rezistenţele echivalente din circuitele discretizate au aceleaşi valori la toţi paşii de timp, deci matricea G rămâne constantă în tot timpul simulării. În consecinţă matricea G poate fi generată o singură dată şi factorizată LU, urmând ca la fiecare pas de timp să se determine potenţialele prin rezolvarea sistemului $L\,U\,V = i_s$. În acest fel, timpul global de calcul capătă ordinul $O(N^3/3 + n\,N^2)$.

Analiza erorilor

Erorile soluţiei numerice se datoresc următoarelor cauze:

- erori inerente, datorate impreciziei datelor de intrare;
- erori de trunchiere, datorate aproximării derivatelor prin diferenţe finite;
- erori de rotunjire, datorate reprezentării finite a numerelor reale apărute la rezolvarea sistemului liniar.

Prin utilizarea metodei implicite, în exprimarea derivatelor, se obţine o stabilitate numerică bună.

În figura 8.30 sunt reprezentate grafic soluţiile de regim tranzitoriu ale circuitului liniar din figura 8.29, obţinute în urma aplicării algoritmului prezentat anterior.

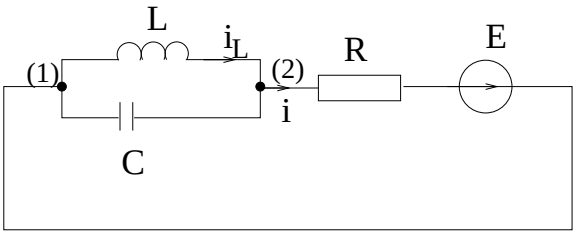


Figura 8.29: Circuit liniar în regim tranzitoriu

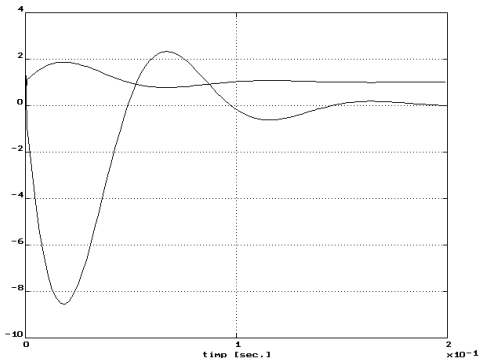


Figura 8.30: Soluţia circuitului 8.29: variaţia în timp a curentului i şi a potenţialului v_1

Prin combinarea algoritmului de rezolvare a circuitelor în regim tranzitoriu cu cel de rezolvare a circuitelor neliniare (prezentat în paragraful ???) se obţine un algoritm care permite rezolvarea circuitelor neliniare în regim tranzitoriu. Aplicarea acestui algoritm la rezolvarea unui circuit de tip redresor monoalternanţă cu filtraj capacitiv (figura 8.31a) se obţin formele de variaţie ale curentului prin rezistor şi tensiuni la bornele sarcinii prezentate în figura 8.32.

Variaţia aceloraşi mărimi pentru circuitul redresor fără filtraj (figura 8.31b), este prezentată în figura 8.33.

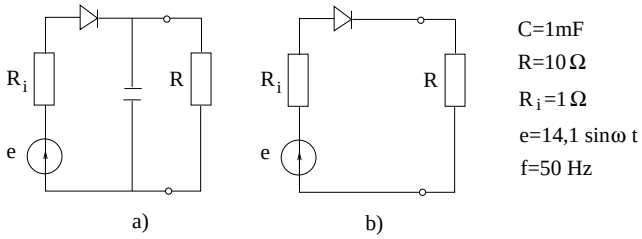


Figura 8.31: Circuite redresoare monoalternanţă: a) cu filtraj capacitiv; b) fără filtraj .

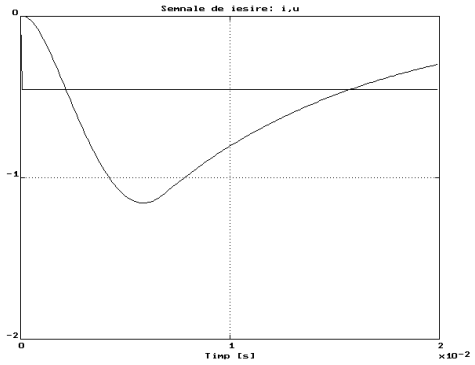


Figura 8.32: Soluţia numerică a circuitului 8.31a

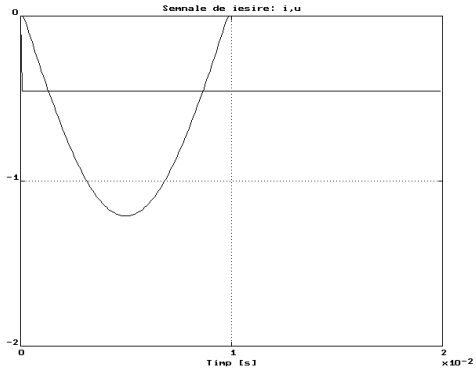


Figura 8.33: Soluţia numerică a circuitului 8.31b

Anexa A

Bibliografie

[1]

[2] ***. *System / 360 Scientific Subroutine Package*. IBM.

[3] Jr A. N. Wilson, editor. *Nonlinear Networks: Theory and analysis*. IEEE Press, NY, 1975.

[4] R. K. Brayton, F. G. Gustavson şi G. D. Hachtel. A new efficient algoritm for solving differential-algebraic systems using implicit backward differentiation formulas. *Proc. of the IEEE*, vol. 60, pp. 98–108, 1972.

[5] R. P. Brent. On the Davidenko-Bronin methods for solving simultaneous nonlinear equations. *IBM J. Res. Develop.*, pp. 434–436, iul. 1972.

[6] C. G. Broyden. A class for solving nonlinear simultaneous equations. *Math. Comp.*, vol. 19, pp. 577–593, 1965.

[7] C. G. Broyden. A new solving nonlinear simultaneous equations ???? *Comput. J.*, vol. 12, pp. 94–99, feb. 1969.

[8] D. A. Calahan. Numerical considerations for implementation of a nonlinear transient analisys program. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.

[9] D. A. Calahan. *Computer-Aided Network Design*. Mc.Graw-Hill, New York, 1972.

[10] B. H. Carnahan, H. A. Luther şi J. O. Wilkes. *Applied Numerical Methods*. John Wiley, NJ, 1971.

[11] K. S. Chao, D. K. Liu şi C. E. Pan. A systematic search method for obtaining multiple solutions of simultaneous nonlinear equation. *IEEE Trans. on CAS*, vol. 22, nr. 9, pp. 748–752, sept. 1975.

[12] M. J. Chien. Searching for multiple solutions of nonlinear systems. *IEEE Trans. on CAS*, vol. 26, nr. 10, pp. 817–827, oct. 1979.

[13] M. J. Chien şi E. Kuh. Solving nonlinear resistive neworks using piecewise-linear analysis and simplicial subdivision. *IEEE Trans. on CAS*, vol. 24, nr. 6, pp. 305–317, iun. 1977.

[14] I. O. Chua şi N. N. Wang. A new approach to overcome the overflow problem in computer-aided analysis of nonlinear resistive circuits. *J. Of Circuit Theory and Applications*, vol. 3, pp. 261–284, 1975.

[15] L. O. Chua şi Y. F. Lem ??? Global homeomorfism of vector-valued functions. *Journal of Mathematical Analysis and Applications*, vol. 39, pp. 600–634, sept. 1972.

[16] L. O. Chua. *Introduction to nonlinear network theory*. McGraw-Hill, NY, 1969.

[17] L. O. Chua. Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks. *IEEE Trans. on CT*, vol. 18, nr. 1, pp. 73–85, ian 1971.

[18] L. O. Chua şi A. C. Deng. Canonical piecewise-linear modelling. *IEEE Trans. on CAS*, vol. 33, nr. 5, pp. 511–525, mai 1986.

[19] L. O. Chua şi A. C. Deng. Canonical piecewise-linear representation. *IEEE Trans. on CAS*, vol. 35, nr. 1, pp. 101–111, ian 1988.

[20] L. O. Chua şi P. M. Lin. *Computer-aided analysis of electronic circuits*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[21] H. Crowder şi P. Wolfe. Linear convergence of the conjugate gradient method. *IBM J. Res. Develop.*, , nr. 7, pp. 431–433, iul. 1972.

[22] W. S. Dorn şi D. D. McCracken. *Metode numerice cu programe în Fortran IV*. Editura Tehnică, Bucureşti, 1976. Traducere din limba engleză.

[23] R. Fletcher şi M. J. D. Powell. A rapidly convergent descendent method for minimization. *Comput. J.*, , nr. 6, pp. 163–168, 1963.

[24] T. Fujisawa, E. Kuh şi T. Ohtsuki. A sparse matrix method for analysis of piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 19, nr. 6, pp. 571–584, nov. 1972.

[25] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[26] C. W. Gear. Simultaneous numerical solutions of differential algebraic equations. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.

[27] P. E. Gill şi W. Murry. Algortihms for the solution of the nonlinear least-square problems. *SIAM J. Numer. Anal.*, vol. 15, nr. 5, pp. 977–992, oct. 1978.

[28] Jr. H. F. Bronin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations*, , nr. IBM J. Res. Develop., pp. 504–522, sept. 1972.

[29] G. D. Hachel, R. K. Brayton şi F. G. Gustavson. The sparse tableau approach to network analisys and design. *IEEE Trans. on CT*, vol. 18, nr. 1, ian 1971.

[30] A. Halanay. *Teoria calitativă a ecuaţiilor diferenţiale*. Ed. Academiei, Bucureşti, 1963.

[31] A. J. Jimenez şi S. W. Director. New families of algorithms for solving nonlinear circuit equations. *IEEE Trans. on CAS*, vol. 25, nr. 1, pp. 1–7, ian 1978.

[32] J. Katzenelson. An algorithm for solving nonlinear resistive networks. *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, oct. 1965.

[33] S. M. Lee şi K. S. Choo. Multiple solutions of piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 30, nr. 2, pp. 84–89, feb. 1983.

[34] W. Liniger. Multistep and one-leg methods for implicit mixed differential and algebraic systems. *IEEE Trans. on CAS*, vol. 26, nr. 9, sept. 1979.

[35] W. Liniger şi F. Odeh. A-stable, accurate averaging of multistep methods for stiff differential equations. *IBM J. of Res. and Develop.*, , nr. 7, iul. 1972.

[36] A. R. Newton şi A. Sangiovanni-Vincentelli. Relaxation-based electrical simulation. *IEEE Trans. on CAD*, vol. 3, nr. 4, pp. 3308–3331, oct. 1984.

[37] T. Ohtsuki şi N. Hoshida. Dc analysis of nonlinear neworks based on generalized piecewise-linear characterization. *IEEE Trans. on CT*, vol. 18, nr. 1, pp. 146–152, ian 1971.

[38] J. M. Ortega şi W. C. Rheinholdt. *Iterative solution of nonlinear equations in several variables*. Academic Press, NY, 1970.

[39] V. C. Prased şi V. P. Prakash. Homeomorphic piecewise-linear resistive networks. *IEEE Trans. on CAS*, vol. 35, nr. 2, pp. 251–253, feb. 1988.

[40] J. R. Rice. *Numerical Methods, Software and Analysis. IMSL Reference Edition*. McGraw Hill, NY, 1983.

[41] A. Rolston şi H. S. Wilf. *Mathematical Methods for Digital Computers*. John Wiley, New York, 1967.

[42] A. E. Ruehli. *Circuit Analysis Simulation and Design*. Elseviers Science Pub., 1986.

[43] G. Russel. *Computer aided tools for VLSI system design*. Peter Peregrims, Ltd., 1987.

[44] I. W. Sandbery. Theorems on the computation of the transient response of nonlinear networks containing transistors and diodes. *Bell Syst. Techn. Journ.*, vol. 49, pp. 1739–1776, oct. 1970.

[45] M. A. F. Schwarz. *Computer-Aided Design of Microelectronic Circuits and Systems*. Academic Press, London, 1987.

[46] K. N. Staton şi N. Talkdar. New integration algorithms for transient stability studies. *IEEE Trans. on PAS*, vol. 89, nr. 56, mai 1970.

[47] A. H. Stroud. *Numerical Quadrature and Solutions of Ordinary Differential Equations*. Springer Verlag, New York, 1974.

[48] B. Thomas. The Runge-Kutta methods. *Byte*, , nr. 4, apr. 1986.

[49] R. Varga. *Matrix iterative analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.

[50] F. F. Wu şi C. A. Desoer. Global inverse function theorem. *IEEE Trans. on CT*, , nr. 3, pp. 199–201, mar. 1972.

[51] D. A. Zein. Solution of a set of nonlinear algebraic equations for general-purpose CAD programs. *EEE Circuits and Devices Magazine*, pp. 7–20, sept. 1985.